

Lifting Bounded-Collusion Security to Full Security in Pairing-Based Encryption Schemes

Roy Stracovsky*

Brent Waters†

David J. Wu‡

Abstract

Notions like identity-based encryption (IBE) and attribute-based encryption (ABE) augment public-key encryption to provide fine-grained access control to encrypted data. In these settings, there is a single master public key that anyone can encrypt to. Users in turn possess different secret keys that determine which ciphertexts they can decrypt. The standard, or full, security notion for these schemes stipulates that a ciphertext computationally hides the message if the adversary does not possess a key that is allowed to decrypt. A relaxed notion of security that is often much easier to achieve is bounded-collusion security, where we only require security against adversaries that have an a priori bounded number of keys (and the scheme parameters can grow with this bound). For instance, it is known that vanilla public-key encryption implies bounded-collusion IBE (and ABE) whereas there is a black-box separation between public-key encryption and fully secure IBE.

In this work, we develop a general methodology to upgrade a pairing-based encryption scheme with bounded-collusion security into one that is fully secure. We illustrate our methodology by applying it to two main applications: batched IBE and multi-authority ABE. First, in the case of batched IBE, we obtain the first pairing-based construction of batched IBE where there are no tag restrictions (in the generic bilinear group model). All previous pairing-based batched IBE schemes require attaching a tag to ciphertexts and secret keys, and moreover, restrict the adversary to an a priori bounded number of secret keys for each tag. Second, we use our techniques to obtain a multi-authority ABE scheme that supports conjunction policies in the generic bilinear group model. This is the first pairing-based scheme that makes fully black-box use of the group. Previous pairing-based multi-authority ABE schemes require a random oracle whose output is a group element (which implicitly assumes the underlying group supports a procedure for explainable oblivious sampling).

*Georgia Institute of Technology. Email: rstracovsky3@gatech.edu. Part of this work was done while visiting UT Austin.

†UT Austin and NTT Research. Email: bwaters@cs.utexas.edu.

‡UT Austin. Email: dwu4@cs.utexas.edu.

Contents

1	Introduction	3
1.1	Technical Overview	4
2	Preliminaries	11
3	Tagless Batched Identity-Based Encryption	13
3.1	Batched IBE Definitions	13
3.2	Batched IBE Construction	14
4	Multi-Authority Attribute-Based Encryption without Random Oracles	37
4.1	Multi-Authority ABE Definitions	37
4.2	Multi-Authority ABE Construction	39
A	Identity-Based Encryption	73
A.1	IBE Definitions	74
A.2	IBE Construction	75

1 Introduction

Classic public-key encryption [DH76, RSA78] has an “all-or-nothing” guarantee where users who hold the secret key are able to decrypt ciphertexts and recover the message, while users who do not have the secret key cannot do anything with the ciphertext. Over the course of the last 25 years, the cryptography community has shown how to extend public-key encryption to support more fine-grained access to encrypted data. This led to notions like identity-based encryption (IBE) [Sha84, BF01, Coc01], attribute-based encryption (ABE) [SW05, GPSW06], and ultimately, functional encryption [BW07, BSW11, O’N10]. For example, in IBE, each secret key sk_{id} is additionally associated with an identity id . Using the master public key mpk , anyone can encrypt a message m with respect to a particular identity id' , with the property that only the user who holds a secret key for the *same* identity can decrypt.

Collusion resistance in advanced encryption. A common feature in these more sophisticated public-key encryption schemes is that there are many secret keys associated with the same master public key mpk . In the example of IBE, we can have a secret key for each identity. The most natural security notion (for the case of IBE) would stipulate that even if the adversary has secret keys for any polynomial number of identities $id_1, \dots, id_Q$, a ciphertext encrypted with respect to a new identity $id^* \notin \{id_1, \dots, id_Q\}$ should still computationally hide the message. Namely, the adversary should not be able to mix and match different unauthorized keys to learn information about the encrypted message. We say that such a scheme is *fully collusion resistant*, or more simply, *fully secure*. We say the encryption scheme satisfies *bounded-collusion resistance* if security is only required to hold against adversaries that obtain *at most* Q secret keys, where the collusion bound Q is a priori fixed, and the scheme parameters are allowed to depend on Q .

Constructing schemes with bounded-collusion security is much easier than constructing schemes with full security. For instance, the works of [SS10, GVW12] show how to construct bounded-collusion-secure functional encryption schemes for general circuits just from the minimal assumption of public-key encryption. Since functional encryption generalizes notions like IBE and ABE, their work also gives bounded-collusion-secure IBE and ABE from vanilla public-key encryption. On the flip side, there are black-box separations for building fully collusion-resistant IBE from public-key encryption [BPR⁺08]. The apparent gap between schemes with bounded-collusion security and those with full security remains even when we bring in algebraic assumptions. We illustrate this with a few examples below:

- First, consider the very first construction of IBE from bilinear maps by Boneh and Franklin [BF01]. Security of this construction relied critically on the random oracle model. However, if we only consider bounded-collusion resistance against collusions of size at most Q , then we could replace the random oracle in the Boneh-Franklin scheme with a $(Q + 1)$ -wise independent hash function and prove security of the *same* scheme in the *plain* model [HK04]. The random oracle is used to achieve full security in this context.
- In the more general setting of batched identity-based encryption [AFP25], where secret keys are now associated with a *set* of identities and can be used to decrypt ciphertexts associated with any identity belonging to the set, existing constructions based on bilinear maps are only secure against bounded collusions [AFP25, CGPW25, FPTX25, GWWW26, Pol26].¹ Strictly speaking, existing constructions mitigate the bounded-collusion limitation by working in a tag-based setting where secret keys and ciphertexts are additionally associated with a tag τ , and decryption is only possible when the tag associated with the secret key and the ciphertext match. In the security game, the adversary is restricted to obtaining at most Q secret keys per tag. Essentially, these schemes move the bounded-collusion restriction to be per-tag rather than for the scheme as a whole. It is currently open how to construct batched IBE without tags from bilinear maps (even in the generic group model). Such a scheme would be analogous to a fully collusion-resistant scheme.
- Finally, turning to multi-authority attribute-based encryption, all existing pairing-based constructions [LW11, RW15, OT20, DKW23a, DKW23b, DP23] rely on the random oracle. However, similar to the case of the Boneh-Franklin IBE, if we restrict our attention to bounded-collusion security, then we can replace the random oracle in a construction like [LW11] with a $(Q + 1)$ -wise independent hash function and prove security in the plain model.

¹Batched IBE is commonly used to build batched threshold encryption. Many recent works have also focused on directly building batched threshold decryption (cf. [BFOQ25, ABD⁺26, BNRT26, BCF⁺26, Pol26, ADG⁺26]). However, batched threshold encryption does not imply batched IBE.

This work. In this work, we introduce a new technique that transforms a bounded-collusion-resistant encryption scheme based on bilinear maps into a fully secure scheme. In our approach, we start with an encryption scheme that is secure against collusions of size Q . Intuitively, this means that to break security, the adversary must combine key material from at least $Q + 1$ different secret keys. Our approach is to design a new *algebraic* mechanism with the property that once an adversary tries to combine a sufficiently-large number of keys (e.g., more than Q), then the resulting key is defective and can no longer decrypt. Observe that this yields a fully secure scheme. The Q -collusion-resistance property of the underlying scheme protects against adversarial strategies involving at most Q keys, while our algebraic mechanism protects against adversarial strategies that involve more than Q keys. To formalize this intuition, we resort to the generic bilinear group model [Sho97, BB04b]. We illustrate the flexibility of our approach by applying it to each of the scenarios described above. This yields the following implications:

- **Identity-based encryption.** Applying our technique to the bounded-collusion Boneh-Franklin IBE scheme in the plain model, we obtain a fully secure IBE scheme in the generic bilinear group model. This approach does not subsume existing results and is intended to serve as an illustrative example of our approach. Indeed, there are already numerous routes to argue full security of IBE in the plain model [BB04a, Wat05, Gen06, Wat09, LW10].
- **Batched identity-based encryption.** Applying our technique to the tag-based batched IBE scheme from [GWWW26], we obtain the first pairing-based batched IBE scheme *without* any tag-based restriction. Security again holds in the generic bilinear group model. All previous pairing-based batched IBE schemes (including those analyzed in the generic group model) imposed a tag-based restriction and limited the number of keys that could be given out for each tag.
- **Multi-authority ABE.** Applying our technique to the multi-authority ABE scheme from [LW11], we obtain a fully secure multi-authority ABE scheme that supports conjunction policies in the generic bilinear group model. We note that this result is *incomparable* to previous pairing-based multi-authority ABE schemes [LW11, RW15, OT20, DKW23a, DKW23b, DP23]. All of these schemes rely on a *random oracle* whose output is a group element. This means the constructions are implicitly assuming the underlying group supports an (explainable) oblivious sampleability property where it is possible to sample a uniform random group element without revealing its discrete log. For this reason, these constructions require some non-black-box use of the group in order to be instantiated. An appealing property of our construction is that it is fully black box in the use of the group and indeed, it can be instantiated entirely in the generic bilinear group model. Finally, we do note that our current construction is limited to conjunction policies whereas prior pairing-based constructions in the random oracle model could support general monotone Boolean formulas.

1.1 Technical Overview

We start by providing an overview of our approach to lift a bounded-collusion-secure encryption scheme into a fully secure scheme. To illustrate our techniques, we show how to apply it to a plain-model variant of the Boneh-Franklin IBE scheme with bounded-collusion security. In this case, the final implication we derive is not of interest (fully secure IBE in the generic group model), but the example will be very helpful for illustrating our main ideas. We start by fixing some notation for pairing groups and then recalling the structure of the Boneh-Franklin IBE scheme [BF01].

- **Notation:** Throughout this work, we use asymmetric pairing groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with prime order p . Let $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ be generators and $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be an efficiently-computable non-degenerate bilinear map. We use implicit notation to denote group elements: namely, we write $[x]_1 := g_1^x$, $[x]_2 := g_2^x$, and $[x]_T := e(g_1, g_2)^x$.
- **Master public key:** The master public key in the Boneh-Franklin IBE scheme is $\text{mpk} = (H, [\alpha]_1)$, where $H: \mathbb{Z}_p \rightarrow \mathbb{G}_2$ is a hash function. The master secret key $\text{msk} = \alpha$. Here, we take the identity space to be $\mathbb{Z}_p$.
- **Key-generation:** A secret key for an identity $\text{id} \in \mathbb{Z}_p$ is $\text{sk}_{\text{id}} = \alpha \cdot [h_{\text{id}}]_2 = [\alpha h_{\text{id}}]_2$ where $[h_{\text{id}}]_2 = H(\text{id})$.
- **Encryption:** An encryption of a message $[m]_T$ with respect to an identity $\text{id} \in \mathbb{Z}_p$ is the pair

$$\text{ct} = (s[\alpha]_1 \cdot [h_{\text{id}}]_2 + [m]_T, [s]_1) = ([s\alpha h_{\text{id}} + m]_T, [s]_1).$$

- **Decryption:** To decrypt using $sk_{id} = [\alpha h_{id}]_2$, simply compute

$$\underbrace{[s\alpha h_{id} + m]_T}_{ct} - \underbrace{[s]_1}_{ct} \cdot \underbrace{[\alpha h_{id}]_2}_{sk_{id}} = [m]_T.$$

Boneh and Franklin show that when the hash function H is modeled as a random oracle, the above scheme is fully secure under the decisional bilinear Diffie-Hellman (DBDH) problem. This proof critically relies on the ability to *program* the random oracle and “plant” the DBDH challenge on the challenge identity. As shown in the work of Heng and Kurosawa [HK04], it is possible to prove bounded-collusion security against collusions of size Q in the *plain* model by replacing the hash function H with a $(Q+1)$ -wise independent hash function. Concretely, we would take H to define a polynomial of degree Q and publish the exponents $[u_0]_2, [u_1]_2, \dots, [u_Q]_2$ in the master public key. Now we define

$$H(x) := [u_0]_2 + x \cdot [u_1]_2 + \dots + x^Q \cdot [u_Q]_2 = [u_0 + \sum_{i \in [Q]} u_i x^i]_2 \in \mathbb{G}_2. \quad (1.1)$$

It is not difficult to show that this version of the scheme is *statically* secure against adversaries that make at most Q key-generation queries in the *plain* model (and under the decisional bilinear Diffie-Hellman assumption). Here, static security means that the adversary has to commit to its key-generation queries at the beginning of the security game.

This scheme is completely broken as soon as the adversary obtains more than Q secret keys. This is because the hash function $H: \mathbb{Z}_p \rightarrow \mathbb{G}_2$ is a degree- Q polynomial. Thus, given the secret keys $[\alpha h_{id_1}]_2, \dots, [\alpha h_{id_{Q+1}}]_2$ for $Q+1$ distinct identities, the adversary can use Lagrange interpolation in the exponent to obtain $[\alpha h_{id^*}]_2$ for *any* identity id^* of its choosing. This allows the adversary to generate decryption keys for any identity, which trivially breaks security.

Preventing collusions. To carry out the collusion attack described above, the adversary needs to take a linear combination of $Q+1$ distinct secret keys. Our idea to prevent such collusions is as follows:

- First, we associate a collection of L slots with the ciphertext and the secret key.
- For each slot, the ciphertext will either contain a well-formed helper decryption term or a corrupted term. These terms are needed for decryption. We let L_{ct} denote the number of well-formed helper terms and $L - L_{ct}$ denote the number of corrupted terms. We write $S_{ct} \subseteq [L]$ to denote the set of slots containing well-formed helper decryption terms.
- Each secret key (for an identity id) will demand helper terms for a subset $S_{sk} \subseteq [L]$ of L_{sk} slots. Decryption is successful only if the ciphertext provides a well-formed helper term for each slot $i \in S_{sk}$, and will fail to decrypt if any slot $i \in S_{sk}$ contains a corrupted term. In other words, decryption succeeds whenever $S_{sk} \subseteq S_{ct}$.

The basic intuition underlying correctness and security is as follows:

- For correctness, we require that $S_{sk} \subseteq S_{ct}$ with probability at least $1 - \epsilon$ for some inverse polynomial ϵ . Note that this yields a scheme with ϵ correctness error, but we can reduce the correctness error to a negligible function via standard parallel repetition.
- Suppose an adversary takes a linear combination of $Q+1$ (or more) different secret keys (associated with identities $id_1, \dots, id_{Q+1}$) to break security. The idea now is that the combined key would now demand helper decryption components from $S_{sk_1} \cup \dots \cup S_{sk_{Q+1}}$. If we choose the number of helper terms L_{sk} required by each key to be sufficiently large, we can ensure that with all but negligible probability, it will be the case that $\bigcup_{j \in [Q+1]} S_{sk_j} \not\subseteq S_{ct}$.
- Finally, if an adversary tries a strategy that involves at most Q different secret keys, we should still appeal to the bounded-collusion resistance of the base scheme to argue security.

We illustrate the general idea underlying our slot mechanism in Fig. 1 and describe the algebraic mechanism below (in the context of the Boneh-Franklin IBE scheme):

- **Public parameters:** The setup algorithm additionally samples slot exponents $\gamma_1, \dots, \gamma_L \xleftarrow{R} \mathbb{Z}_p$. The master public key is now $mpk = (H, [\alpha]_1, \{\{\gamma_i\}_1\}_{i \in [L]})$ and the master secret key is $msk = (\alpha, \{\gamma_i\}_{i \in [L]})$. Here, we instantiate the hash function H with the concrete function from Eq. (1.1).



(a) A single secret key individually demands a small number of slots, so with high probability, $S_{sk} \subseteq S_{ct}$ and decryption succeeds.

(b) If an adversary tries to combine multiple secret keys together, decryption succeeds only if the ciphertext provides well-formed helper terms for each slot demanded by one of the underlying keys. In this case, we set the parameters so that with overwhelming probability $\bigcup_j S_{sk_j} \not\subseteq S_{ct}$.

Figure 1: An illustration of the slot mechanism we introduce to defend against collusion attacks. In this example, we consider a scheme with $L = 10$ slots. Each ciphertext has a set S_{ct} of $L_{ct} = 8$ slots containing well-formed helper terms and the remaining slots are corrupted (denoted by an “x”). Each secret key for an identity id demands $L_{sk} = 2$ well-formed helper terms (denoted by a •). Decryption succeeds if all of the slots $S_{sk} \subseteq [L]$ demanded by the secret key are well-formed (i.e., when $S_{sk} \subseteq S_{ct}$).

- **Key-generation:** To generate a key for an identity id , the key-generation algorithm samples a random set $S_{sk} \subseteq [L]$ of size L_{sk} together with fresh blinding factors $\beta_{i,id} \xleftarrow{R} \mathbb{Z}_p$ for each $i \in S_{sk}$. It then masks the Boneh-Franklin key $[ah_{id}]_2$ using the slots in S_{sk} :

$$sk_{id} = ([ah_{id} + \sum_{i \in S_{sk}} \gamma_i \beta_{i,id}]_2, \{[\beta_{i,id}]_2\}_{i \in S_{sk}}).$$

- **Encryption:** To encrypt a message $[m]_T$ to an identity id , the encryption algorithm sets $[c_1]_T = [sah_{id} + m]_T$ and $[c_2]_1 = [s]_1$ as before. Then, it samples a random set $S_{ct} \subseteq [L]$ of size L_{ct} and defines $[c_{3,i}]_1 = [s\gamma_i]_1$ if $i \in S_{ct}$ and $[c_{3,i}]_1 \xleftarrow{R} \mathbb{G}_1$ if $i \notin S_{ct}$. The ciphertext is then

$$ct = ([c_1]_T, [c_2]_1, \{[c_{3,i}]_1\}_{i \in [L]}) = ([sah_{id} + m]_T, [s]_1, \{[c_{3,i}]_1\}_{i \in [L]}).$$

When $c_{3,i} = s\gamma_i$, we say the helper decryption term is well-formed and when $c_{3,i}$ is random, we say it is corrupted.

- **Decryption:** The decryption algorithm proceeds as before, except it must now additionally remove the mask $\sum_{i \in S_{sk}} \gamma_i \beta_{i,id}$. It does so by pairing the slot components of the ciphertext with the blinding factors in the secret key:

$$\begin{aligned} & [c_1]_T - [c_2]_1 \cdot [ah_{id} + \sum_{i \in S_{sk}} \gamma_i \beta_{i,id}]_2 + \sum_{i \in S_{sk}} [c_{3,i}]_1 \cdot [\beta_{i,id}]_2 \\ &= [sah_{id} + m]_T - [sah_{id} + \sum_{i \in S_{sk}} s\gamma_i \beta_{i,id}]_T + \sum_{i \in S_{sk}} [c_{3,i} \beta_{i,id}]_T \\ &= [m + \sum_{i \in S_{sk} \setminus S_{ct}} (r_i - s\gamma_i) \beta_{i,id}]_T, \end{aligned}$$

where we write $[c_{3,i}]_1 = [r_i]_1$ for all $i \notin S_{ct}$. When $S_{sk} \subseteq S_{ct}$, then $S_{sk} \setminus S_{ct} = \emptyset$, and this is precisely $[m]_T$. Conversely, if $S_{sk} \not\subseteq S_{ct}$, then the decryption relation essentially produces a random group element; this is the crux of our security analysis.

In our analysis, we set $L = Q = \lambda$, $L_{sk} = \lambda^{1/3}$, and $L_{ct} = \lambda - \lambda^{1/3}$. This ensures that $\Pr[S_{sk} \not\subseteq S_{ct}] \leq \lambda^{-1/3}$, which ensures correctness with inverse polynomial error (which we can reduce via parallel repetition). Moreover, we can show that if we consider the union $\bigcup_{j \in [Q+1]} S_{sk_j}$ of $Q + 1$ such sets $S_{sk_1}, \dots, S_{sk_{Q+1}}$, then with overwhelming probability $\bigcup_{j \in [Q+1]} S_{sk_j} \not\subseteq S_{ct}$. We refer to [Appendix A](#) for the formal description of the IBE scheme.

Analyzing security in the generic group model. To reason about the security of our construction, we work in the generic bilinear group model [Sho97, BB04b]. In this model, we replace group elements with random handles, and the only way the adversary can evaluate the group operation (or pairing) is through querying the generic group oracle. Thus, every group element that the adversary can construct can be expressed as either a linear function or a quadratic function of the group elements it has available. Working in the generic bilinear group model allows us to precisely reason about the adversarial strategies the adversary is employing. For simplicity, we work in the model where the generic group oracle associates a fresh random handle with each generic group query and there is an explicit *zero-test* oracle for testing whether a handle is an encoding of 0 or not. Any lower bound in this model can be translated to one in Shoup’s model (where we associate a unique handle with every group element) with quadratic overhead.

In our security proof (see the proof of [Theorem A.10](#)), we show how to simulate the view of the adversary in the IBE security game without knowledge of the message $[m]_T$. We proceed via a hybrid argument over generic group oracle queries that the adversary makes in the security game. For the generic group queries, it suffices to analyze the distribution of the zero-test queries since the other oracles always output fresh encodings on each query. Our simulation strategy now goes as follows:

- Whenever the adversary makes a key-generation query on an identity id , the simulator samples the set of required indices S_{sk} and responds with S_{sk} along with a fresh set of handles representing the group elements in sk_{id} . The simulator associates each of these group elements with the identity id .
- When the adversary makes a zero-test query to the generic group oracle, the simulator inspects the variables associated with the query. If the query does not involve $[m]_T$, then its value is a function of quantities that the simulator sampled itself. In this case, the simulator can answer the query without knowing $[m]_T$. It thus suffices to only consider queries that depend on $[c_1]_T$. For these queries, we argue that the adversary cannot obtain the cancellations needed for the zero-test oracle to output “zero,” irrespective of $[m]_T$. The simulator can thus answer “non-zero” without knowing $[m]_T$.

The crux of the argument is to show that every zero-test query that involves $[c_1]_T = [sah_{id^*} + m]_T$ always evaluates to a non-zero value. We consider two cases based on the number of secret keys the query combines:

- **At most Q keys:** To cancel out $[c_1]_T$, the adversary must construct a linear combination of group elements that evaluates to $[sah_{id^*}]_T$. The only way to produce monomials containing both s and α is by pairing together $[c_2]_1 = [s]_1$ with a secret key component $[\alpha h_{id} + \sum_{i \in S_{sk}} \gamma_i \beta_{i,id}]_2$. By $(Q + 1)$ -wise independence of H , the value of h_{id^*} is independent of $h_{id_1}, \dots, h_{id_Q}$ as long as $id^* \notin \{id_1, \dots, id_Q\}$. Since the adversary cannot make a key-generation query on the challenge identity, we conclude that the query polynomial is a non-zero polynomial (so answering “non-zero” is correct, independent of the value of $[m]_T$).
- **More than Q keys:** Suppose the adversary pairs together $[s]_1$ with $[\alpha h_{id_j} + \sum_{i \in S_{sk_j}} \gamma_i \beta_{i,id_j}]_2$ for at least $Q + 1$ different identities $id_1, \dots, id_{Q+1}$. This accumulates many monomials of the form $s\gamma_i\beta_{i,id_j}$. The only way to cancel these out is to pair the slot component $[c_{3,i}]_1 = [s\gamma_i]_1$ with the blinding term $[\beta_{i,id_j}]_2$ for every slot i in the union $\bigcup_{j \in [Q+1]} S_{sk_j}$. This is possible only if $\bigcup_{j \in [Q+1]} S_{sk_j} \subseteq S_{ct}$. Now, if we can argue that the adversary’s view up until this point is *independent* of S_{ct} , then over the randomness of S_{ct} , with overwhelming probability, it will be the case that $\bigcup_{j \in [Q+1]} S_{sk_j} \not\subseteq S_{ct}$. Again, the reduction can answer such queries with “non-zero,” irrespective of the value of $[m]_T$.

This step of the analysis is contingent on the adversary’s view being independent of the set S_{ct} . This is why the reduction can *defer* the sampling of S_{ct} until *after* the adversary has committed to its query, and thus, the set $\bigcup_{j \in [Q+1]} S_{sk_j}$. To ensure this, we modify our hybrid invariant to show that each query the adversary makes can be simulated without knowledge of either $[m]_T$ or S_{ct} . It turns out that essentially the same analysis as above suffices to show this more general invariant. We also note that security critically relies on the set S_{ct} being hidden. If S_{ct} is public, the adversary can repeatedly query for individual keys and reject any key that is incompatible with the ciphertext set S_{ct} . Then, the adversary can carry out the standard collusion attack using only the keys that are “good.” Thus, it is important that S_{ct} is hidden from the adversary, and moreover, that the adversary cannot learn information about S_{ct} unless it considers a query that involves more than Q different keys. This is the invariant we establish in our formal proof (see the proof of [Theorem A.10](#)).

Extending to batched identity-based encryption. The basic template described above extends naturally to obtain collusion-resistant versions of other pairing-based encryption schemes. For instance, in a batched IBE scheme [AFP25], each secret key is associated with a *set* I of up to B identities and can be used to decrypt ciphertexts encrypted to any identity $\text{id} \in I$. As mentioned earlier, existing pairing-based batched IBE schemes [AFP25, CGPW25, FPTX25, GWWW26, Pol26] are only secure in the bounded-collusion model (or consider a tag-based model where they limit the number of secret keys that can be given out per tag). By applying our approach to the scheme of [GWWW26], we obtain a batched IBE scheme that supports issuing an arbitrary number of decryption keys (without any tag-based restrictions). To illustrate our approach, we first recall the scheme from [GWWW26] that supports issuing Q secret keys. We describe a slimmed-down version of the scheme *without* tags.

- **Public parameters:** The master public key is $\text{mpk} = ([\alpha]_{\mathbb{T}}, [\mathbf{w}]_1, [\tau\mathbf{w}]_1, \{[\tau^j]_2\}_{j \in [0, B]})$, where $\alpha, \tau \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and $\mathbf{w} \xleftarrow{\mathbb{R}} \mathbb{Z}_p^Q$. The master secret key is $\text{msk} = (\alpha, \tau, \mathbf{w})$.
- **Key-generation:** A secret key for a set I of identities is $\text{sk}_I = (\mathbf{y}_I, [\alpha + \mathbf{y}_I^{\top} \mathbf{w} \cdot F_I(\tau)]_2)$, where $\mathbf{y}_I \xleftarrow{\mathbb{R}} \mathbb{Z}_p^Q$ and $F_I(X) = \prod_{\text{id} \in I} (X - \text{id})$.
- **Encryption:** An encryption of a message $[m]_{\mathbb{T}}$ with respect to an identity id is the tuple

$$\text{ct} = ([s\alpha + m]_{\mathbb{T}}, [s]_1, [s(\tau - \text{id})\mathbf{w}]_1).$$

- **Decryption:** To decrypt using sk_I when $\text{id} \in I$, write $F_I(X) = (X - \text{id}) \cdot F_{I \setminus \{\text{id}\}}(X)$. Then compute $[F_{I \setminus \{\text{id}\}}(\tau)]_2$ from $[1]_2, [\tau]_2, \dots, [\tau^B]_2$. To decrypt, observe that

$$[s]_1 \cdot [\alpha + \mathbf{y}_I^{\top} \mathbf{w} \cdot F_I(\tau)]_2 - \mathbf{y}_I^{\top} [s(\tau - \text{id})\mathbf{w}]_1 \cdot [F_{I \setminus \{\text{id}\}}(\tau)]_2 = [s\alpha]_{\mathbb{T}}.$$

This is sufficient to recover $[m]_{\mathbb{T}}$ from $[s\alpha + m]_{\mathbb{T}}$.

In this scheme, the vector $\mathbf{w} \in \mathbb{Z}_p^Q$ plays the role of the $(Q + 1)$ -wise independent hash in the Boneh-Franklin IBE scheme. Each secret key reveals a single linear function $\mathbf{y}_I^{\top} \mathbf{w}$ of $\mathbf{w}$. With overwhelming probability, the vectors $\mathbf{y}_I$ associated with any Q keys are linearly independent, so the adversary must collect more than Q keys before they become linearly dependent. To obtain a fully collusion-resistant scheme, we introduce the slots exactly as before:

- First we add the slot exponents $\gamma_1, \dots, \gamma_L$ to the master public key.
- We add a masking term to each secret key. Specifically, the secret key sk_I for a set I is now

$$\text{sk}_I = (\mathbf{y}_I, [\alpha + \mathbf{y}_I^{\top} \mathbf{w} \cdot F_I(\tau) + \sum_{i \in S_{\text{sk}}} \gamma_i \beta_{i,I}]_2, \{[\beta_{i,I}]_2\}_{i \in S_{\text{sk}}}).$$

- During encryption, we first sample the set S_{ct} and the ciphertext contains $[c_{4,i}]_1 = [s\gamma_i]_1$ for all $i \in S_{\text{ct}}$ and $[c_{4,i}]_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1$ for all $i \in [L] \setminus S_{\text{ct}}$.

It is easy to see that as long as $S_{\text{sk}} \subseteq S_{\text{ct}}$, decryption succeeds. The security analysis in the generic bilinear group model follows a similar argument as that used to analyze the Boneh-Franklin IBE scheme. We give the formal construction and analysis of our batched IBE scheme in [Section 3](#).

Extending to multi-authority attribute-based encryption. As a final application, we show how to apply our techniques to the [LW11] multi-authority ABE scheme. First, in standard (ciphertext-policy) ABE [SW05, GPSW06], secret keys are associated with attributes whereas ciphertexts are associated with an access policy. Decryption succeeds whenever the attributes associated with the key satisfy the policy associated with the ciphertext. In this setting, there is a single authority that is issuing keys for different attribute sets. In multi-authority ABE, different (and independent) authorities manage different subsets of attributes, and policies can be defined over the set of attributes associated with multiple authorities. To tie the secret keys for different authorities together, we associate a global identifier gid with each user. Existing pairing-based multi-authority ABE schemes [LW11, RW15, OT20, DKW23a, DKW23b, DP23] all hash the gid to derive a group element, and the security analysis models this hash function as a random oracle.

However, the work of [LW11] also notes that the hash function could have been replaced by a $(Q+1)$ -wise independent hash function, and the scheme could be proven secure against adversaries that possess at most Q secret keys. Our techniques described above can be used to lift this scheme to a fully collusion-resistant scheme in the generic bilinear group model. This yields a construction that is fully black box in the use of the group. We start by recalling the structure of the [LW11] scheme, for the simplified setting where each authority manages a single attribute:

- **Global parameters:** The global parameters consist of the description of the hash function $H: \mathbb{Z}_p \rightarrow \mathbb{G}_2$, where H is the $(Q+1)$ -wise independent hash function from Eq. (1.1). Here, we model the user identifiers as elements of $\mathbb{Z}_p$.
- **Authority setup:** Each authority a samples $\alpha_a, y_a \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and publishes a public key $\text{mpk}_a = ([\alpha_a]_{\mathbb{T}}, [y_a]_1)$.
- **Key-generation:** A secret key from authority a for a user with identifier gid is $\text{sk}_{a,\text{gid}} = [\alpha_a + y_a h_{\text{gid}}]_2$, where $[h_{\text{gid}}]_2 = H(\text{gid})$.
- **Encryption:** The [LW11] scheme supports any policy with a linear secret sharing scheme. To encrypt a message $[m]_{\mathbb{T}}$ with respect to a policy associated with authorities $a_1, \dots, a_\ell$ where the public key for a_i is $\text{mpk}_{a_i} = ([\alpha_{a_i}]_{\mathbb{T}}, [y_{a_i}]_1)$, the encryption algorithm proceeds as follows:
 - Sample a random $s \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and secret share s according to the policy. Let $v_i \in \mathbb{Z}_p$ be the share associated with authority a_i . For ease of exposition, we assume each authority appears in the policy exactly once.
 - Similarly, secret share 0 according to the policy and let $w_i \in \mathbb{Z}_p$ be the share associated with a_i .
 - Finally, for each $i \in [\ell]$, sample a blinding factor $r_i \xleftarrow{\mathbb{R}} \mathbb{Z}_p$.

The ciphertext is then

$$\text{ct} = ([s + m]_{\mathbb{T}}, \{[v_i + r_i \alpha_{a_i}]_{\mathbb{T}}, [r_i]_1, [r_i y_{a_i} + w_i]_1\}_{i \in [\ell]}).$$

- **Decryption:** Suppose a user with global identifier gid holds secret keys from a set of authorities $S \subseteq [\ell]$ that satisfies the policy. For each $i \in S$, the user pairs the ciphertext components associated with authority a_i with its secret key $\text{sk}_{a_i,\text{gid}}$ and $[h_{\text{gid}}]_2$:

$$[v_i + r_i \alpha_{a_i}]_{\mathbb{T}} - [r_i]_1 \cdot [\alpha_{a_i} + y_{a_i} h_{\text{gid}}]_2 + [r_i y_{a_i} + w_i]_1 \cdot [h_{\text{gid}}]_2 = [v_i + w_i h_{\text{gid}}]_{\mathbb{T}}.$$

Since v_i is a secret share of s while w_i is a secret share of 0, and S satisfies the policy, the user can take a linear combination of $[v_i + w_i h_{\text{gid}}]_{\mathbb{T}}$ to obtain $[s]_{\mathbb{T}}$, which allows them to recover the message from $[s + m]_{\mathbb{T}}$.

In this scheme, the hash function H plays a similar role as that in the Boneh-Franklin IBE scheme. Each secret key reveals a single evaluation of H , so H is under-determined until the adversary collects more than Q keys for different user identifiers. To obtain a fully collusion-resistant scheme, we introduce the slots exactly as before, except each authority has its own set of slot exponents. We describe the scheme below:

- First, each authority a samples slot exponents $\gamma_{1,a}, \dots, \gamma_{L,a} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and adds $\{[\gamma_{i,a}]_1\}_{i \in [L]}$ to its public key mpk_a .
- We add a masking term to each secret key. Specifically, the secret key that authority a issues to a user gid is now

$$\text{sk}_{a,\text{gid}} = ([\alpha_a + y_a h_{\text{gid}} + \sum_{i \in S_{a,\text{gid}}} \gamma_{i,a} \beta_{i,a,\text{gid}}]_2, \{[\beta_{i,a,\text{gid}}]_2\}_{i \in S_{a,\text{gid}}}),$$

where $S_{a,\text{gid}} \subseteq [L]$ is the set of slots the user demands (for authority a).

- During encryption, we sample a set $S_{\text{ct},i}$ for each authority a_i associated with the policy, and the ciphertext additionally contains a slot component $[c_{i,j}]_1$ for each $j \in [L]$, where $[c_{i,j}]_1 = [r_i \gamma_{j,a_i}]_1$ if $j \in S_{\text{ct},i}$ and $[c_{i,j}]_1 = [r'_{i,j}]_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1$ is uniform if $j \in [L] \setminus S_{\text{ct},i}$.

Whenever $S_{a_i, \text{gid}} \subseteq S_{\text{ct}, i}$ and the user has a secret key for authority a_i , the blinding terms cancel and the user recovers the shares $[v_i + w_i h_{\text{gid}}]_{\mathbb{T}}$, exactly as before. Since $S_{a_i, \text{gid}} \subseteq S_{\text{ct}, i}$ must hold for *all* authorities, we need to additionally assume an upper bound on the number of authorities that can appear in each policy (and then allow the parameters to scale with the bound M). Note that we need a different set $S_{\text{ct}, i}$ for each authority to ensure that the sets associated with *honest* authorities remain hidden even if some of the other authorities in the ciphertext are corrupted. Using the same set for all authorities violates this property and breaks our security analysis.

While the construction described above satisfies correctness for any policy that can be described by a linear secret sharing scheme, to prove security, we need to further restrict to linear secret sharing schemes with a stronger *share uniformity* property. This property requires that the joint distribution of the shares associated with any unauthorized set of attributes be independent and uniformly random. To see why, consider a user with identifier gid who holds a secret key

$$\text{sk}_{a_i, \text{gid}} = ([\delta_{a_i, \text{gid}}]_2, \{[\beta_{j, a_i, \text{gid}}]_2\}_{j \in S_{a_i, \text{gid}}}), \quad \text{where} \quad \delta_{a_i, \text{gid}} = \alpha_{a_i} + y_{a_i} h_{\text{gid}} + \sum_{j \in S_{a_i, \text{gid}}} \gamma_{j, a_i} \beta_{j, a_i, \text{gid}},$$

from authority a_i . Suppose the user uses $\text{sk}_{a_i, \text{gid}}$ to recover the i^{th} share $[v_i + w_i h_{\text{gid}}]_{\mathbb{T}}$. In this case, the user computes

$$\begin{aligned} & [v_i + r_i \alpha_{a_i}]_{\mathbb{T}} - [r_i]_1 \cdot [\delta_{a_i, \text{gid}}]_2 + [r_i y_{a_i} + w_i]_1 \cdot [h_{\text{gid}}]_2 + \sum_{j \in S_{a_i, \text{gid}}} [c_{i, j}]_1 \cdot [\beta_{j, a_i, \text{gid}}]_2 \\ &= \left[v_i + w_i h_{\text{gid}} + \underbrace{\sum_{j \in S_{a_i, \text{gid}} \setminus S_{\text{ct}, i}} (r'_{i, j} - r_i \gamma_{j, a_i}) \beta_{j, a_i, \text{gid}}}_{e_i} \right]_{\mathbb{T}}. \end{aligned}$$

We can view this process as recovering a “noisy” version of the share $[v_i + w_i h_{\text{gid}}]_{\mathbb{T}}$, where the noise is $[e_i]_{\mathbb{T}}$. In most cases, $S_{a_i, \text{gid}} \subseteq S_{\text{ct}, i}$, and $e_i = 0$, but with noticeable probability, $e_i \neq 0$. Suppose the user was able to tell whether $e_i = 0$ or $e_i \neq 0$. This would reveal whether $S_{a_i, \text{gid}} \subseteq S_{\text{ct}, i}$ or $S_{a_i, \text{gid}} \not\subseteq S_{\text{ct}, i}$. Since $S_{a_i, \text{gid}}$ is known to the user, this now reveals information about the set $S_{\text{ct}, i}$, which leads to an attack on the scheme. If the shares for any unauthorized set of attributes were independent and uniform, then we could rely on $v_i + w_i h_{\text{gid}}$ to mask e_i . However, if the shares v_i and w_i were not uniform, then we cannot make this argument. Indeed, in this setting, there is an explicit attack on the scheme.

Concretely, consider the policy $a_1 \wedge (a_2 \vee a_3)$. Suppose (v_1, v_2, v_3) and (w_1, w_2, w_3) are secret shares of s and 0 under this policy, respectively. Under the standard secret sharing scheme for monotone Boolean formulas [BL88, LW11], we would set $v_2 = v_3$ and $w_2 = w_3$. Now, consider an adversary that obtains secret keys $\text{sk}_{a_2, \text{gid}}$ and $\text{sk}_{a_3, \text{gid}}$ from authorities a_2 and a_3 , respectively. Note that this is an unauthorized collection of keys. Using the above decryption relation, the adversary can now compute $[v_2 + w_2 h_{\text{gid}} + e_2]_{\mathbb{T}}$ and $[v_3 + w_3 h_{\text{gid}} + e_3]_{\mathbb{T}}$. Since $v_2 = v_3$ and $w_2 = w_3$, the adversary now learns $[e_2 - e_3]_{\mathbb{T}}$. If this is 0 , then with overwhelming probability, this means $e_2 = 0$ and $e_3 = 0$ and thus, $S_{a_2, \text{gid}} \subseteq S_{\text{ct}, 2}$ and $S_{a_3, \text{gid}} \subseteq S_{\text{ct}, 3}$. The adversary now has the ability to test whether the keys $\text{sk}_{a_2, \text{gid}}$ and $\text{sk}_{a_3, \text{gid}}$ are “compatible” with the challenge ciphertext or not. By assembling more than Q compatible keys, it is able to interpolate the $(Q + 1)$ -wise independent hash function H and craft a key for a_2 and a_3 with respect to a new user identifier gid^* of its choosing. Then, by requesting a key from a_1 for identifier gid^* , the adversary now breaks security.

For this reason, we are limited to only supporting linear secret sharing schemes with the stronger share uniformity property. This property is equivalent to asking that the rows of the share-generating matrix associated with any unauthorized set of attributes be linearly independent. This captures policy classes like conjunctions and, more broadly, DNF formulas. We note that this issue of “zeroizing” attacks due to releasing noisy shares frequently occurs in the setting of lattice-based constructions [DKW21, CW26a, AGY26a, CW26b, AGY26b, GY26], and it is for this reason that all of these aforementioned schemes are restricted to conjunction and DNF policies. We give our full construction and analysis of our multi-authority ABE scheme in Section 4.

Open problem: proving security in the plain model. In this work, we analyze the security of our technique in the generic bilinear group model. Working in the generic bilinear group model provides a clean way to design defenses against the two main classes of adversarial strategies: (1) bounded-collusion security handles adversarial strategies that try to combine a small number of keys; and (2) our slot randomization mechanism protects against adversarial strategies that combine a large number of keys. An interesting challenge is to try to prove security of our construction in the plain model from a concrete assumption. Such a result would give plain-model constructions of fully secure batched IBE as well as multi-authority ABE for conjunction policies. Neither of these primitives is currently known from pairings in the plain model.

2 Preliminaries

In this work, we let λ denote the security parameter. For integers $a, b \in \mathbb{Z}$ with $a \leq b$, we write $[a, b] := \{a, \dots, b\}$ and $[a] := [1, a]$ when $a \geq 1$. We write $\mathbb{Z}_p$ to denote the ring of integers modulo a prime p . We let $\{0, 1\}^k$ denote bit-strings of length k and assume they can be implicitly interpreted as elements of $[0, 2^k - 1]$. For a set S , we write $\{x_i\}_{i \in S}$ to denote the set of *pairs* $\{(i, x_i) : i \in S\}$ (i.e., $\{x_i\}_{i \in S}$ specifies both the set S and the association between each index $i \in S$ and the corresponding element x_i). When S is finite, we write $x \stackrel{\mathcal{R}}{\leftarrow} S$ to denote a random sample from S . If not otherwise stated, sampling is always uniform. A table $\mathcal{T}$ consists of entries (or mappings) $a \mapsto b$. To retrieve an entry, we write $b = \mathcal{T}(a)$.

We write boldface lowercase letters (e.g., $\mathbf{u}, \mathbf{v}$) for vectors and boldface uppercase letters (e.g., $\mathbf{A}, \mathbf{B}$) for matrices. We use non-boldface letters to denote components (e.g., $\mathbf{u} = (u_1, \dots, u_n)$). For a vector $\mathbf{u}$, we let $(\mathbf{u})_i$ denote the i^{th} component (i.e., $(\mathbf{u})_i = u_i$). For a polynomial $F(X)$ of degree at most n over $\mathbb{Z}_p$, we define its coefficient vector $\mathbf{f} = (f_1, f_2, \dots, f_{n+1}) \in \mathbb{Z}_p^{n+1}$ such that $F(X) = \sum_{i \in [n+1]} f_i X^{i-1}$.

We write $\text{poly}(\lambda)$ to denote a positive-valued function that is upper bounded by a fixed polynomial in λ (i.e., there exist $c, d \in \mathbb{N}$ such that $\text{poly}(\lambda) \leq \lambda^c$ for all $\lambda \geq d$). We write $\text{negl}(\lambda)$ to denote a positive-valued function that is $o(\lambda^{-c})$ for all constants $c \in \mathbb{N}$ (i.e., for all $c \in \mathbb{N}$, there exists some $d \in \mathbb{N}$ such that $\text{negl}(\lambda) \leq \lambda^{-c}$ for all $\lambda \geq d$). In prose, we may abuse notation and write a growth class (e.g., $\omega(\lambda)$) to refer to a *fixed* function in that growth class. We say an algorithm is efficient if it runs in probabilistic polynomial time in the length of its input.

Prime-order pairing groups. Throughout this work, we use asymmetric prime-order pairing groups. The following presentation is taken nearly verbatim from [SWW26].

Definition 2.1 (Prime-Order Pairing Group). An asymmetric prime-order pairing group consists of an efficient algorithm GroupGen that takes as input the security parameter 1^λ and outputs a description of a pairing group $\tilde{\mathcal{G}} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \bar{g}_1, \bar{g}_2, e)$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of prime order p satisfying $2^\lambda < p = 2^{\Theta(\lambda)}$, $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map, and $\bar{g}_1, \bar{g}_2$ are generators of $\mathbb{G}_1, \mathbb{G}_2$, respectively. We require that the group operations in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are efficiently computable. For ease of exposition, we assume that there is a fixed function $p(\lambda)$ such that for each security parameter λ , $\text{GroupGen}(1^\lambda)$ outputs a group of order $p = p(\lambda)$.

Implicit notation. Throughout this work, we will describe group elements using implicit notation [EHK⁺13]. Specifically, fix a pair of generators g_1 and g_2 for $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. For a matrix $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$, we write $[\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{A}]_T$ to denote $g_1^{\mathbf{A}}, g_2^{\mathbf{A}}, e(g_1, g_2)^{\mathbf{A}}$, respectively, where exponentiation is defined component-wise. For matrices $\mathbf{A}, \mathbf{B}$ with compatible dimensions, we define $[\mathbf{A}]_i + [\mathbf{B}]_i := [\mathbf{A} + \mathbf{B}]_i$ and for a scalar $s \in \mathbb{Z}_p$, we define $s[\mathbf{A}]_i := [s\mathbf{A}]_i$, where $i \in \{1, 2, T\}$. We define $[\mathbf{A}]_1 \cdot [\mathbf{B}]_2 := [\mathbf{AB}]_T$, where the pairing is used to compute the pairwise products between the elements in $[\mathbf{A}]_1$ and $[\mathbf{B}]_2$. We will often work relative to a *random* choice of generators $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ (as opposed to the generators $\bar{g}_1, \bar{g}_2$ output by GroupGen). We denote this by writing $[1]_1 := g_1$ and $[1]_2 := g_2$.

Formal polynomials. Let $\mathcal{V}$ be a set of symbols or formal variables. A formal polynomial f over $\mathcal{V}$ is an element of the polynomial ring $\mathbb{Z}_p[\mathcal{V}]$. For formal polynomials f_1, f_2 , polynomial addition and multiplication are denoted $f_1 + f_2$ and $f_1 \cdot f_2$, respectively. Every formal polynomial f can be written uniquely as $f = c_1 \cdot m_1 + \dots + c_n \cdot m_n$, where $m_1, \dots, m_n$ are distinct monic monomials over $\mathcal{V}$ (i.e., monomials with coefficient 1) and $c_1, \dots, c_n \in \mathbb{Z}_p$ are nonzero. For a monic monomial m over $\mathcal{V}$, we define $\text{coeff}_m(f) = c_i$ if $m = m_i$ for some $i \in [n]$, and $\text{coeff}_m(f) = 0$ otherwise. We say that f has a term m if $\text{coeff}_m(f) \neq 0$ and that a term m has a variable $v \in \mathcal{V}$ if it can be written as the product of v and some monomial m' over $\mathcal{V}$. We say a polynomial f has a variable v if it has a term with variable v .

Generic group model. We formalize generic bilinear groups using a simplified version of [Zim15, Definition 2.10], which was originally defined for generic multilinear maps. We will also make use of the Schwartz-Zippel lemma.

Definition 2.2 (Generic Bilinear Group). An (asymmetric) *generic bilinear group* over a prime p and handle space $\mathcal{H}$ is a tuple of oracles $\text{BGGM} = (\text{BGGM.Encode}, \text{BGGM.Add}, \text{BGGM.Pair}, \text{BGGM.ZeroTest})$ that share an internal table $\mathcal{T}$ and operate as follows:

- **BGGM.Encode**(x, v): On input an exponent $x \in \mathbb{Z}_p$ and an index $v \in \{1, 2, T\}$ (representing the “left” group $\mathbb{G}_1$, the “right” group $\mathbb{G}_2$, or the target group $\mathbb{G}_T$ respectively), the oracle samples a handle $\langle h \rangle \xleftarrow{R} \mathcal{H}$ and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- **BGGM.Add**($\langle h_1 \rangle, \langle h_2 \rangle$): On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if $v_1 \neq v_2$. It then computes $x = x_1 + x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{R} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v_1)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- **BGGM.Pair**($\langle h_1 \rangle, \langle h_2 \rangle$): On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if either $v_1 \neq 1$ or $v_2 \neq 2$. It then computes $x = x_1 x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{R} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, T)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- **BGGM.ZeroTest**($\langle h \rangle$): On input a handle $\langle h \rangle \in \mathcal{H}$, the oracle retrieves $\mathcal{T}(\langle h \rangle) = (x, v)$ and aborts with output “invalid” if the entry is undefined. It then returns “zero” if $x = 0 \in \mathbb{Z}_p$ and “non-zero” otherwise.

Any security experiment in the plain model can be adapted to the generic group model by augmenting the challenger so that it maintains a generic bilinear group BGGM, replaces each group element it sends to the adversary with a handle produced by BGGM.Encode, and allows the adversary to query the BGGM oracles during its query phases. We define these experiments explicitly in our proofs. We say a scheme is secure in the generic group model if it is secure in the corresponding generic group experiment against all efficient adversaries that make polynomially many queries to BGGM.

Lemma 2.3 (Schwartz-Zippel [Sch80, Zip79]). *Fix a prime p and let $f \in \mathbb{Z}_p[x_1, \dots, x_n]$ be an n -variate polynomial of degree at most d that is not identically zero. Then*

$$\Pr[f(x_1, \dots, x_n) = 0 : x_1, \dots, x_n \xleftarrow{R} \mathbb{Z}_p] \leq \frac{d}{p}.$$

Vandermonde matrices. We define a square Vandermonde matrix $\mathbf{V} = V(x_1, x_2, \dots, x_n)$ as

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{pmatrix}.$$

Over $\mathbb{Z}_p$ (or more broadly, any finite field), the matrix $\mathbf{V}$ is invertible if and only if $x_1, x_2, \dots, x_n$ are distinct.

Linear secret sharing. Our multi-authority ABE scheme makes use of linear secret sharing schemes with a linear independence property, both of which are defined below. Our presentation is adapted from [SWW26].

Definition 2.4 (Linear Secret Sharing Scheme [Bei96, adapted]). A linear secret sharing scheme (LSSS) for a set A over $\mathbb{Z}_p$ is a pair $(\mathbf{M}, \rho)$, where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times n}$ is a share-generating matrix and $\rho: [\ell] \rightarrow A$ is a row-labeling function. For a subset $B \subseteq A$, define $X_B = \{x \in [\ell] : \rho(x) \in B\}$ and let $\mathbf{M}_B \in \mathbb{Z}_p^{|X_B| \times n}$ be the matrix formed by taking the subset of rows in $\mathbf{M}$ indexed by X_B . The pair $(\mathbf{M}, \rho)$ defines an access policy P as follows:

- **Authorized sets:** We say a subset $B \subseteq A$ satisfies P if there exists a vector $\omega_B \in \mathbb{Z}_p^{|X_B|}$ such that $\omega_B^\top \mathbf{M}_B = \mathbf{e}_1^\top$, where $\mathbf{e}_1$ is the first elementary basis vector.
- **Unauthorized sets:** We say a subset $B \subseteq A$ does not satisfy P if $\mathbf{e}_1^\top$ is not in the row-span of $\mathbf{M}_B$. Equivalently, there exists a vector $\mathbf{v}^* \in \mathbb{Z}_p^n$ with $v_1^* = 1$ such that $\mathbf{M}_B \mathbf{v}^* = \mathbf{0}$.

To share a value $s \in \mathbb{Z}_p$ with respect to P , sample $v_2, \dots, v_n \xleftarrow{R} \mathbb{Z}_p$ and define the vector $\mathbf{v} = (s, v_2, \dots, v_n)$. Then, $\mathbf{u} = \mathbf{M}\mathbf{v} \in \mathbb{Z}_p^\ell$ is the vector of shares where u_i belongs to party $\rho(i) \in A$ for each $i \in [\ell]$. We say that the size of P is the number of rows ℓ in the matrix $\mathbf{M}$.

Definition 2.5 (Linear Independence [DKW21, adapted]). We say a linear secret sharing scheme $(\mathbf{M}, \rho)$ is linearly independent for unauthorized sets if, for every unauthorized set $B \subseteq A$, the rows of $\mathbf{M}_B$ are linearly independent.

3 Tagless Batched Identity-Based Encryption

Batched identity-based encryption [AFP25] generalizes the notions of batch decryption and identity-based encryption so that a (small) secret key is associated with a *set* of identities and decryption is possible whenever a ciphertext is encrypted to an identity contained within the set. Existing pairing-based schemes [AFP25, CGPW25, FPTX25, GWWW26, Pol26] also require keys and ciphertexts to be associated with a “batch label” and only consider security against adversaries who obtain a single secret key (or more generally up to an a priori bound K keys) for each batch label. In this section, we apply our technique to build a pairing-based batched identity-based encryption scheme that *does not* require batch label tags. We refer to such a scheme as *tagless*.

3.1 Batched IBE Definitions

We start by recalling the syntax and security notions for batched identity-based encryption [AFP25]. Our formulation is for the *tagless* setting.

Definition 3.1 (Batched IBE [AFP25, adapted]). A *batched identity-based encryption* (batched IBE) scheme with identity space $\mathcal{I} = \{\mathcal{I}_\lambda\}_{\lambda \in \mathbb{N}}$ is a tuple of efficient algorithms $\Pi_{\text{BatchIBE}} = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ with the following syntax:

- $\text{Setup}(1^\lambda, 1^B) \rightarrow (\text{mpk}, \text{msk})$: On input the security parameter λ and a batch size bound B , the setup algorithm outputs a master public key mpk and a master secret key msk . We assume that mpk implicitly contains the security parameter 1^λ and a description of the message space $\mathcal{M}_\lambda$ (where $|\mathcal{M}_\lambda| \geq 2$).
- $\text{KeyGen}(\text{msk}, I) \rightarrow \text{sk}_I$: On input the master secret key msk and a set of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$, the key-generation algorithm outputs an identity-batch secret key sk_I associated with I .
- $\text{Encrypt}(\text{mpk}, \text{id}, m) \rightarrow \text{ct}$: On input the master public key mpk , an identity $\text{id} \in \mathcal{I}_\lambda$, and a message $m \in \mathcal{M}_\lambda$, the (randomized) encryption algorithm outputs a ciphertext ct .
- $\text{Decrypt}(\text{mpk}, I, \text{sk}_I, \text{ct}) \rightarrow m$: On input the master public key mpk , a set of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$, an identity-batch secret key sk_I associated with I , and a ciphertext ct , the decryption algorithm outputs a message m .

Definition 3.2 (Correctness). A batched IBE scheme Π_{BatchIBE} is *statistically correct* with error $\varepsilon(\lambda)$ if, for all $\lambda, B \in \mathbb{N}$, all master key-pairs (mpk, msk) in the support of $\text{Setup}(1^\lambda, 1^B)$, all sets of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$, all identity-batch secret keys sk_I in the support of $\text{KeyGen}(\text{msk}, I)$, all identities $\text{id} \in I$, and all messages $m \in \mathcal{M}_\lambda$ (where $\mathcal{M}_\lambda$ is the message space associated with mpk),

$$\Pr [\text{Decrypt}(\text{mpk}, I, \text{sk}_I, \text{ct}) = m : \text{ct} \leftarrow \text{Encrypt}(\text{mpk}, \text{id}, m)] \geq 1 - \varepsilon(\lambda).$$

Remark 3.3 (Weaker Notions of Correctness). Our definition above requires correctness to hold with a fixed probability for all master key-pairs and identity secret keys; that is, the probability is taken solely over the encryption randomness. This is the analog of “almost-all-keys perfect correctness” from [DNR04] (adapted to the setting of batched IBE). One could also consider weaker notions of correctness where this probability is taken over the randomness of key generation, or even both key generation and setup. While valid, these relaxations are meaningfully weaker. In particular, under our definition, we can reduce the decryption failure by encrypting the message multiple times. Under the weaker definitions, there could exist non-functional secret keys (or even a non-functional master key-pair) where decryption is guaranteed to fail irrespective of the encryption randomness.

Remark 3.4 (Negligible Correctness Error). Our definition allows the correctness error $\varepsilon(\lambda)$ to be non-negligible. When $\varepsilon(\lambda) \leq 1/2 - 1/\text{poly}(\lambda)$, we can achieve *negligible correctness error* by performing a $\text{poly}'(\lambda)$ -wise repetition of encryption (without performing multiple setups or generating multiple keys for each user and thus not affecting any

succinctness requirements on keys) where $\text{poly}'(\lambda) = \omega(\log \lambda) \cdot \text{poly}(\lambda)^2$. The decryption algorithm then outputs the majority of the decrypted messages. This form of amplification is possible as the correctness error in [Definition 3.2](#) is taken over the encryption randomness (as opposed to the key-generation or setup randomness); see [Remark 3.3](#).

Definition 3.5 (Succinctness). A batched IBE scheme Π_{BatchIBE} is *succinct* if there exists a universal polynomial $\text{poly}(\cdot)$ such that, for all $\lambda, B \in \mathbb{N}$, all master key-pairs (mpk, msk) in the support of $\text{Setup}(1^\lambda, 1^B)$, and all sets of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$, every identity-batch secret key sk_I in the support of $\text{KeyGen}(\text{msk}, I)$ satisfies $|\text{sk}_I| \leq \text{poly}(\lambda)$ (i.e., $|\text{sk}_I|$ is independent of the batch size bound B).

Definition 3.6 (Semantic Security). For a batched IBE scheme Π_{BatchIBE} , an adversary $\mathcal{A}$, a bit $b \in \{0, 1\}$, and a security parameter $\lambda \in \mathbb{N}$, we define the semantic security experiment as follows:

1. **Setup phase:** The adversary $\mathcal{A}$ sends a batch size bound 1^B to the challenger. In response, the challenger samples $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, 1^B)$ and gives mpk to $\mathcal{A}$.
2. **Pre-challenge query phase:** The adversary $\mathcal{A}$ can now make key-generation queries. On each query, the adversary specifies a set of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$ and the challenger replies with $\text{sk}_I \leftarrow \text{KeyGen}(\text{msk}, I)$.
3. **Challenge phase:** At some point, $\mathcal{A}$ outputs a challenge identity $\text{id}^* \in \mathcal{I}_\lambda$ together with two challenge messages $m_0^*, m_1^* \in \mathcal{M}_\lambda$. The challenger replies with $\text{ct}^* \leftarrow \text{Encrypt}(\text{mpk}, \text{id}^*, m_b^*)$.
4. **Post-challenge query phase:** Now, $\mathcal{A}$ can continue to make key-generation queries. Like the pre-challenge query phase, the challenger replies with $\text{sk}_I \leftarrow \text{KeyGen}(\text{msk}, I)$ when $\mathcal{A}$ queries a set $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$.
5. **Output phase:** At the end of the experiment, $\mathcal{A}$ outputs a bit $b^* \in \{0, 1\}$, which is the output of the experiment.

We say that an adversary $\mathcal{A}$ is admissible if it does not make any key-generation queries on any set of identities I containing the challenge identity id^* . We furthermore say that Π_{BatchIBE} is semantically secure if, for every efficient admissible adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$|\Pr[b^* = 1 : b = 0] - \Pr[b^* = 1 : b = 1]| \leq \text{negl}(\lambda).$$

3.2 Batched IBE Construction

In this section, we describe our construction of tagless batched identity-based encryption. As described in [Section 1.1](#), we augment the batched IBE scheme from [\[GW26\]](#) (which supports giving out $Q = K$ decryption keys per batch label tag) with our slot mechanism (see also [Fig. 1](#)). We give our full construction below:

Construction 3.7 (Tagless Batched Identity-Based Encryption). Let λ be a security parameter and define the slot count $L = L(\lambda)$, the number of slots $L_{\text{sk}} = L_{\text{sk}}(\lambda)$ associated with each secret key, number of slots $L_{\text{ct}} = L_{\text{ct}}(\lambda)$ associated with each ciphertext, the collusion bound $K = K(\lambda)$, and the prime bit-length $P = P(\lambda)$. Let GroupGen be a prime-order asymmetric pairing-group generator. We construct a tagless batched identity-based encryption scheme $\Pi_{\text{BatchIBE}} = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ with identity space $\mathcal{I} = \{\mathcal{I}_\lambda\}_{\lambda \in \mathbb{N}}$ where $\mathcal{I}_\lambda = \{0, 1\}^\lambda$ as follows:

- **Setup** $(1^\lambda, 1^B)$: On input the security parameter λ and batch size bound B , the setup algorithm proceeds as follows:
 - Sample a prime-order asymmetric pairing group $\bar{\mathcal{G}} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \bar{g}_1, \bar{g}_2, e) \leftarrow \text{GroupGen}(1^{P(\lambda)})$ and generators $g_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1 \setminus \{\bar{g}_1^0\}$ and $g_2 \xleftarrow{\mathbb{R}} \mathbb{G}_2 \setminus \{\bar{g}_2^0\}$. Define $[1]_1 := g_1$, $[1]_2 := g_2$, and $\mathcal{G} = (\bar{\mathcal{G}}, [1]_1, [1]_2)$.
 - Sample exponents $\alpha, \tau, w_1, \dots, w_K, \gamma_1, \dots, \gamma_L \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and define $\mathbf{w} = (w_1, \dots, w_K)$ and $\boldsymbol{\tau} = (1, \tau, \dots, \tau^B)$.

Output the master public key $\text{mpk} = (\mathcal{G}, [\alpha]_T, [\mathbf{w}]_1, [\boldsymbol{\tau}]_2, \{[\gamma_i]_1\}_{i \in [L]})$ as well as the master secret key $\text{msk} = (\mathcal{G}, \alpha, \mathbf{w}, \boldsymbol{\tau}, \{\gamma_i\}_{i \in [L]})$. Define the message space to be $\mathcal{M}_\lambda := \mathbb{G}_T$.

- **KeyGen** (msk, I) : On input the master secret key $\text{msk} = (\mathcal{G}, \alpha, \mathbf{w}, \boldsymbol{\tau}, \{\gamma_i\}_{i \in [L]})$ and a set of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$, the key-generation algorithm proceeds as follows:
 - Sample exponents $y_{1,I}, \dots, y_{K,I} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and define $y_I = (y_{1,I}, \dots, y_{K,I})$.

- Sample a set $S_I \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$ and sample exponents $\beta_{i,I} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ for all $i \in S_I$.
- Define the polynomial $F_I(X) := \prod_{\text{id} \in I} (X - \text{id})$ and let $\mathbf{f}_I \in \mathbb{Z}_p^{B+1}$ be its associated coefficient vector.
- Compute $[\delta_I]_2 = [\alpha + \mathbf{y}_I^\top \mathbf{w} \cdot \mathbf{f}_I^\top \boldsymbol{\tau} + \sum_{i \in S_I} \gamma_i \beta_{i,I}]_2$.

Output the identity-batch secret key $\text{sk}_I = (\mathbf{y}_I, [\delta_I]_2, \{[\beta_{i,I}]_2\}_{i \in S_I})$.

- **Encrypt**($\text{mpk}, \text{id}, [m]_\top$): On input the master public key $\text{mpk} = (\mathcal{G}, [\alpha]_\top, [\mathbf{w}]_1, [\tau \mathbf{w}]_1, [\boldsymbol{\tau}]_2, \{[\gamma_i]_1\}_{i \in [L]})$, an identity $\text{id} \in \mathcal{I}_\lambda$, and a message $[m]_\top \in \mathcal{M}_\lambda$, the encryption algorithm proceeds as follows:
 - Sample an exponent $s \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and compute the following ciphertext components:

$$[c_1]_\top = s[\alpha]_\top + [m]_\top \quad , \quad [c_2]_1 = [s]_1 \quad , \quad [c_3]_1 = s([\tau \mathbf{w}]_1 - \text{id}[\mathbf{w}]_1) .$$

- Sample a set $S_{\text{ct}} \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$ and let $[c_{4,i}]_1 = s[\gamma_i]_1$ for $i \in S_{\text{ct}}$ and $[c_{4,i}]_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1$ for $i \in [L] \setminus S_{\text{ct}}$.

Output the ciphertext $\text{ct} = (\text{id}, [c_1]_\top, [c_2]_1, [c_3]_1, \{[c_{4,i}]_1\}_{i \in [L]})$.

- **Decrypt**($\text{mpk}, I, \text{sk}_I, \text{ct}$): On input the master public key $\text{mpk} = (\mathcal{G}, [\alpha]_\top, [\mathbf{w}]_1, [\tau \mathbf{w}]_1, [\boldsymbol{\tau}]_2, \{[\gamma_i]_1\}_{i \in [L]})$, a set of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$, an identity-batch secret key $\text{sk}_I = (\mathbf{y}_I, [\delta_I]_2, \{[\beta_{i,I}]_2\}_{i \in S_I})$, and a ciphertext $\text{ct} = (\text{id}, [c_1]_\top, [c_2]_1, [c_3]_1, \{[c_{4,i}]_1\}_{i \in [L]})$, the decryption algorithm proceeds as follows:
 - Define the polynomial $F_{I \setminus \{\text{id}\}}(X) := \prod_{\text{id}' \in I \setminus \{\text{id}\}} (X - \text{id}')$ and let $\mathbf{f}_{I \setminus \{\text{id}\}} \in \mathbb{Z}_p^{B+1}$ be its associated coefficient vector.
 - Output the following message:

$$[m]_\top = [c_1]_\top - [c_2]_1 \cdot [\delta_I]_2 + \mathbf{y}_I^\top [c_3]_1 \cdot \mathbf{f}_{I \setminus \{\text{id}\}}^\top [\boldsymbol{\tau}]_2 + \sum_{i \in S_I} [c_{4,i}]_1 \cdot [\beta_{i,I}]_2 .$$

Theorem 3.8 (Correctness). *Suppose $P \geq \lambda$. Then [Construction 3.7](#) is statistically correct with error $\epsilon(\lambda) = L_{\text{sk}}(1 - L_{\text{ct}}/L)$.*

Proof. Take any $\lambda, B \in \mathbb{N}$, and any (mpk, msk) in the support of $\text{Setup}(1^\lambda, 1^B)$ where $\text{mpk} = (\mathcal{G}, [\alpha]_\top, [\mathbf{w}]_1, [\tau \mathbf{w}]_1, [\boldsymbol{\tau}]_2, \{[\gamma_i]_1\}_{i \in [L]})$. Next, take any set $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$ and let sk_I be in the support of $\text{KeyGen}(\text{msk}, I)$ where $\text{sk}_I = (\mathbf{y}_I, [\delta_I]_2, \{[\beta_{i,I}]_2\}_{i \in S_I})$ for

$$[\delta_I]_2 = \left[\alpha + \mathbf{y}_I^\top \mathbf{w} \cdot \mathbf{f}_I^\top \boldsymbol{\tau} + \sum_{i \in S_I} \gamma_i \beta_{i,I} \right]_2 .$$

For any $\text{id} \in I$ and $[m]_\top \in \mathcal{M}_\lambda$, let $\text{ct} \leftarrow \text{Encrypt}(\text{mpk}, \text{id}, [m]_\top)$ where $\text{ct} = (\text{id}, [c_1]_\top, [c_2]_1, [c_3]_1, \{[c_{4,i}]_1\}_{i \in [L]})$ for

$$\begin{aligned} [c_1]_\top &= s[\alpha]_\top + [m]_\top = [s\alpha + m]_\top \quad , \\ [c_2]_1 &= [s]_1 \quad , \\ [c_3]_1 &= s([\tau \mathbf{w}]_1 - \text{id}[\mathbf{w}]_1) = [s(\tau - \text{id})\mathbf{w}]_1 \quad , \end{aligned} \quad [c_{4,i}]_1 = \begin{cases} s[\gamma_i]_1 = [s\gamma_i]_1 & i \in S_{\text{ct}} \\ [r_i]_1 & i \notin S_{\text{ct}} \end{cases} ,$$

where for each $i \notin S_{\text{ct}}$, $r_i \in \mathbb{Z}_p$ denotes the discrete logarithm of the uniformly random group element $[c_{4,i}]_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1$ sampled by **Encrypt**. Now consider $[m^*]_\top \leftarrow \text{Decrypt}(\text{mpk}, I, \text{sk}_I, \text{ct})$, and recall that id is a root of the polynomial $F_I(X) = \prod_{\text{id}' \in I} (X - \text{id}')$ associated with the coefficient vector $\mathbf{f}_I$ (since $\text{id} \in I$), but it is not a root of the polynomial $F_{I \setminus \{\text{id}\}}(X) = \prod_{\text{id}' \in I \setminus \{\text{id}\}} (X - \text{id}')$ associated with the coefficient vector $\mathbf{f}_{I \setminus \{\text{id}\}}$ (because $P \geq \lambda$, so the mapping from

identities to exponents is injective). We have that $F_I(\tau) = (\tau - \text{id}) \cdot F_{I \setminus \{\text{id}\}}(\tau)$, and thus

$$\begin{aligned}
[m^*]_{\mathbb{T}} &= [c_1]_{\mathbb{T}} - [c_2]_1 \cdot [\delta_I]_2 + \mathbf{y}_I^{\mathbb{T}} [c_3]_1 \cdot \mathbf{f}_{I \setminus \{\text{id}\}}^{\mathbb{T}} [\tau]_2 + \sum_{i \in S_I} [c_{4,i}]_1 \cdot [\beta_{i,I}]_2 \\
&= [s\alpha + m]_{\mathbb{T}} - \left[s\alpha + s\mathbf{y}_I^{\mathbb{T}} \mathbf{w} \cdot \mathbf{f}_I^{\mathbb{T}} \tau + s \sum_{i \in S_I} \gamma_i \beta_{i,I} \right]_{\mathbb{T}} + \left[s(\tau - \text{id}) \mathbf{y}_I^{\mathbb{T}} \mathbf{w} \cdot \mathbf{f}_{I \setminus \{\text{id}\}}^{\mathbb{T}} \tau \right]_{\mathbb{T}} + \sum_{i \in S_I \cap S_{\text{ct}}} [s\gamma_i \beta_{i,I}]_{\mathbb{T}} + \sum_{i \in S_I \setminus S_{\text{ct}}} [r_i \beta_{i,I}]_{\mathbb{T}} \\
&= \left[s\alpha + m - s\alpha - s\mathbf{y}_I^{\mathbb{T}} \mathbf{w} \cdot F_I(\tau) - s \sum_{i \in S_I} \gamma_i \beta_{i,I} + s\mathbf{y}_I^{\mathbb{T}} \mathbf{w} \cdot F_{I \setminus \{\text{id}\}}(\tau) + \sum_{i \in S_I \cap S_{\text{ct}}} s\gamma_i \beta_{i,I} + \sum_{i \in S_I \setminus S_{\text{ct}}} r_i \beta_{i,I} \right]_{\mathbb{T}} \\
&= \left[s\alpha + m - s\alpha - s\mathbf{y}_I^{\mathbb{T}} \mathbf{w} \cdot F_I(\tau) - \sum_{i \in S_I} s\gamma_i \beta_{i,I} + s\mathbf{y}_I^{\mathbb{T}} \mathbf{w} \cdot F_I(\tau) + \sum_{i \in S_I \cap S_{\text{ct}}} s\gamma_i \beta_{i,I} + \sum_{i \in S_I \setminus S_{\text{ct}}} r_i \beta_{i,I} \right]_{\mathbb{T}} \\
&= \left[m - \sum_{i \in S_I \setminus S_{\text{ct}}} s\gamma_i \beta_{i,I} + \sum_{i \in S_I \setminus S_{\text{ct}}} r_i \beta_{i,I} \right]_{\mathbb{T}} \\
&= \left[m + \sum_{i \in S_I \setminus S_{\text{ct}}} (r_i - s\gamma_i) \beta_{i,I} \right]_{\mathbb{T}}.
\end{aligned}$$

Observe that if $S_I \subseteq S_{\text{ct}}$, then $S_I \setminus S_{\text{ct}} = \emptyset$ and consequently the blinding term $\sum_{i \in S_I \setminus S_{\text{ct}}} (r_i - s\gamma_i) \beta_{i,I} = 0$. Hence, in this event, $[m^*]_{\mathbb{T}} = [m]_{\mathbb{T}}$ and decryption succeeds. It remains to bound the probability that $S_I \not\subseteq S_{\text{ct}}$. Since S_{ct} is sampled independently of S_I , we can appeal to the union bound and conclude

$$\Pr[S_I \not\subseteq S_{\text{ct}}] \leq \sum_{i \in S_I} \Pr[i \notin S_{\text{ct}}] = L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L} \right). \quad \square$$

Theorem 3.9 (Succinctness). *Suppose $L = \text{poly}_1(\lambda)$ and $K = \text{poly}_2(\lambda)$, $L_{\text{sk}} \leq L$, and $P = \text{poly}_3(\lambda)$ and $2^{P(\lambda)} < p = 2^{\Theta(P(\lambda))}$. Then [Construction 3.7](#) is succinct.*

Proof. Take any $\lambda, B \in \mathbb{N}$, and any (mpk, msk) in the support of $\text{Setup}(1^\lambda, 1^B)$. Next, take any set of identities $I \subseteq \mathcal{I}_\lambda$ with $0 < |I| \leq B$ and let sk_I be in the support of $\text{KeyGen}(\text{msk}, I)$ where $\text{sk}_I = (\mathbf{y}_I, [\delta_I]_2, \{[\beta_{i,I}]_2\}_{i \in S_I})$. Observe that any sk_I contains K exponents in $\mathbb{Z}_p$, $L_{\text{sk}} + 1$ group elements in $\mathbb{G}_2$, and L_{sk} indices.

Now recall that GroupGen is called on input $1^{P(\lambda)}$ and outputs primes p such that $2^{P(\lambda)} < p = 2^{\Theta(P(\lambda))}$. It follows that there exists some $c, d \in \mathbb{N}$ such that $\log p \leq c \cdot P(\lambda)$ for all $\lambda \geq d$. Let $\text{poly}_4(x)$ denote the size of group elements in $\mathbb{G}_2$ given x -bit exponents. We have that $|\text{sk}_I|$ is bounded by

$$\begin{aligned}
|\text{sk}_I| &\leq K \cdot c \cdot \text{poly}_3(\lambda) + (L_{\text{sk}} + 1) \cdot \text{poly}_4(c \cdot \text{poly}_3(\lambda)) + L_{\text{sk}} \cdot \log L \\
&\leq \text{poly}_2(\lambda) \cdot c \cdot \text{poly}_3(\lambda) + (\text{poly}_1(\lambda) + 1) \cdot \text{poly}_4(c \cdot \text{poly}_3(\lambda)) + \text{poly}_1(\lambda) \cdot \log(\text{poly}_1(\lambda)) \\
&= \text{poly}(\lambda),
\end{aligned}$$

by standard polynomial closure under addition, multiplication, and composition. $\square$

Theorem 3.10 (Security). *Suppose $L = \text{poly}_1(\lambda)$ and $K = \text{poly}_2(\lambda)$, $L_{\text{sk}} = \omega(\log \lambda)$ and $K \cdot L_{\text{sk}} \geq \kappa \cdot L$ for some constant $\kappa \geq 2$, $L - L_{\text{ct}} = \omega(\log \lambda)$, and $P = \omega(K \cdot \log \lambda) \geq \lambda$ and $2^{P(\lambda)} < p = 2^{\Theta(P(\lambda))}$. Then [Construction 3.7](#) is semantically secure in the generic bilinear group model of order p .*

Proof. Consider the following hybrids parameterized by a bit b and a security parameter λ and, implicitly, an efficient admissible adversary $\mathcal{A}$. We use [green](#) to denote changes between hybrids.

- **Hybrid $\text{Hyb}_0^{(b)}(\lambda)$:** This hybrid is equivalent to an admissible adversary $\mathcal{A}$ playing the semantic security game against [Construction 3.7](#) where the underlying bilinear group is modeled as a *generic* bilinear group

of order p as per Definition 2.2. We let q_{KG} , q_{BGGM} , and q_{ZT} denote upper bounds on the respective number of key-generation, generic bilinear group operation (i.e., encode, add, and pair oracles), and zero-test oracle queries $\mathcal{A}$ makes. The game proceeds as follows:

- **Setup phase:** The challenger, upon receiving a batch size bound 1^B from the adversary $\mathcal{A}$, performs the setup procedure as follows:
 - * The challenger samples a random prime $2^{P(\lambda)} < p = 2^{\Theta(P(\lambda))}$, sets the BGGM handle space $\mathcal{H} = \{0, 1\}^\lambda$, and also initializes the table $\mathcal{T}$ which maps handles to exponents.
 - * Next, it samples exponents $\alpha, \tau, w_1, \dots, w_K, \gamma_1, \dots, \gamma_L \xleftarrow{\mathbb{R}} \mathbb{Z}_p$.
 - * For each $i \in [K]$, it executes both $\text{BGGM.Encode}(w_i, 1)$ and $\text{BGGM.Encode}(\tau w_i, 1)$ to produce handles $\langle w_i \rangle$ and $\langle \tau w_i \rangle$ respectively. Furthermore, for each $j \in [0, B]$, it executes $\text{BGGM.Encode}(\tau^j, 2)$ to produce a handle $\langle \tau^j \rangle$. Let $\mathbf{w} = (w_1, \dots, w_K)$ and $\boldsymbol{\tau} = (1, \tau, \dots, \tau^B)$.
 - * Finally, it executes $\text{BGGM.Encode}(\alpha, T)$ to produce a handle $\langle \alpha \rangle$ and, for each $i \in [L]$, it executes $\text{BGGM.Encode}(\gamma_i, 1)$ to produce a handle $\langle \gamma_i \rangle$.

The challenger sends $\text{mpk} = (p, \langle \alpha \rangle, \{\langle w_i \rangle\}_{i \in [K]}, \{\langle \tau w_i \rangle\}_{i \in [K]}, \{\langle \tau^j \rangle\}_{j \in [0, B]}, \{\langle \gamma_i \rangle\}_{i \in [L]})$ to $\mathcal{A}$.

- **Pre-challenge key-generation phase:** The adversary $\mathcal{A}$ may make key-generation queries on sets of its choosing. Let I_ξ denote the set of identities queried on the ξ^{th} key-generation query, recalling that it must hold that $0 < |I_\xi| \leq B$. The challenger generates an identity-batch secret key as follows:
 - * The challenger samples exponents $y_{1,\xi}, \dots, y_{K,\xi} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and defines $\mathbf{y}_\xi = (y_{1,\xi}, \dots, y_{K,\xi})$.
 - * Next, it samples a set $S_\xi \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$ and, for each $i \in S_\xi$, samples an exponent $\beta_{i,\xi} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and executes $\text{BGGM.Encode}(\beta_{i,\xi}, 2)$ to produce a handle $\langle \beta_{i,\xi} \rangle$.
 - * Finally, it defines the polynomial $F_\xi(X) := \prod_{\text{id} \in I_\xi} (X - \text{id})$ with coefficient vector $\mathbf{f}_\xi \in \mathbb{Z}_p^{B+1}$, computes $\delta_\xi = \alpha + \mathbf{y}_\xi^\top \mathbf{w} \cdot \mathbf{f}_\xi^\top \boldsymbol{\tau} + \sum_{i \in S_\xi} \gamma_i \beta_{i,\xi}$, and executes $\text{BGGM.Encode}(\delta_\xi, 2)$ to produce a handle $\langle \delta_\xi \rangle$.

The challenger sends the identity-batch secret key $\text{sk}_\xi = (\mathbf{y}_\xi, \langle \delta_\xi \rangle, \{\langle \beta_{i,\xi} \rangle\}_{i \in S_\xi})$ to the adversary $\mathcal{A}$. During this phase, $\mathcal{A}$ may also query the generic group oracles. We reiterate their functionality below:

- * $\text{BGGM.Encode}(x, v)$: On input an exponent $x \in \mathbb{Z}_p$ and an index $v \in \{1, 2, T\}$, the oracle samples a handle $\langle h \rangle \xleftarrow{\mathbb{R}} \mathcal{H}$ and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- * $\text{BGGM.Add}(\langle h_1 \rangle, \langle h_2 \rangle)$: On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if $v_1 \neq v_2$. It then computes $x = x_1 + x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{\mathbb{R}} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v_1)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- * $\text{BGGM.Pair}(\langle h_1 \rangle, \langle h_2 \rangle)$: On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if either $v_1 \neq 1$ or $v_2 \neq 2$. It then computes $x = x_1 x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{\mathbb{R}} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, T)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- * $\text{BGGM.ZeroTest}(\langle h \rangle)$: On input a handle $\langle h \rangle \in \mathcal{H}$, the oracle retrieves $\mathcal{T}(\langle h \rangle) = (x, v)$ and aborts with output “invalid” if the entry is undefined. It returns “zero” if $x = 0 \in \mathbb{Z}_p$ and “non-zero” otherwise.
- **Challenge phase:** In this phase, the adversary $\mathcal{A}$ outputs a challenge identity id^* (which is not contained in any set queried during the pre-challenge phase as $\mathcal{A}$ is admissible) together with handles $\langle m_0^* \rangle, \langle m_1^* \rangle$ for $\mathcal{A}$ ’s challenge messages such that $\mathcal{T}(\langle m_0^* \rangle) = (m_0^*, T)$ and $\mathcal{T}(\langle m_1^* \rangle) = (m_1^*, T)$. Namely, both messages are in the target group. The challenger now produces the challenge ciphertext ct^* encrypting $\langle m_b^* \rangle$ under id^* as follows:
 - * The challenger samples an exponent $s \xleftarrow{\mathbb{R}} \mathbb{Z}_p$, retrieves the entry $\mathcal{T}(\langle m_b^* \rangle) = (m_b^*, T)$, and then runs $\text{BGGM.Encode}(s\alpha + m_b^*, T)$ to produce a handle $\langle c_1^* \rangle$.
 - * Next, it executes $\text{BGGM.Encode}(s, 1)$ to produce a handle $\langle c_2^* \rangle$.
 - * Then, for each $i \in [K]$, it executes $\text{BGGM.Encode}(s(\tau - \text{id}^*)w_i, 1)$ to produce a handle $\langle c_{3,i}^* \rangle$.

- * Finally, it samples a set $S^* \xleftarrow{R} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. For each $i \in [L]$, it computes $c_{4,i}^* = sy_i$ if $i \in S^*$ and samples $c_{4,i}^* \xleftarrow{R} \mathbb{Z}_p$ otherwise, and it executes $\text{BGGM.Encode}(c_{4,i}^*, 1)$ to produce a handle $\langle c_{4,i}^* \rangle$. The challenger sends the challenge ciphertext $\text{ct}^* = (\text{id}^*, \langle c_1^* \rangle, \langle c_2^* \rangle, \{\langle c_{3,i}^* \rangle\}_{i \in [K]}, \{\langle c_{4,i}^* \rangle\}_{i \in [L]})$ to $\mathcal{A}$.
- **Post-challenge key-generation phase:** The (admissible) adversary $\mathcal{A}$ may again make key-generation queries on sets of its choosing, provided those sets do not contain the challenge identity id^* . The challenger generates identity-batch secret keys using the same procedure as the pre-challenge key-generation phase. During this phase, $\mathcal{A}$ may again make encode, add, pair, and zero-test oracle queries as specified above.
- **Output phase:** The adversary outputs a bit b^* which is the output of the game.
- **Hybrid $\text{Hyb}_1^{(b)}(\lambda)$:** This hybrid is equivalent to hybrid $\text{Hyb}_0^{(b)}$, except we add the following halting condition:
 - **Key-generation phases:** If there exists a collection of K or fewer vectors $\mathbf{y}_{\xi_1}, \dots, \mathbf{y}_{\xi_{K'}} \in \mathbb{Z}_p^K$ (i.e., where $K' \leq K$) sampled during key-generation that are linearly dependent, halt with output 0.
- **Hybrid $\text{Hyb}_2^{(b)}(\lambda)$:** This hybrid is equivalent to hybrid $\text{Hyb}_1^{(b)}$, except we add the following halting condition:
 - **Key-generation queries:** If there exists a collection of K sets $S_{\xi_1}, \dots, S_{\xi_K} \subseteq [L]$ sampled during distinct key-generation queries (i.e., $\xi_1, \dots, \xi_K$ are all different) such that $|\bigcup_{i \in [K]} S_{\xi_i}| \leq L/\kappa$, halt with output 0.
- **Hybrid $\text{Hyb}_3^{(b)}(\lambda)$:** This hybrid is equivalent to hybrid $\text{Hyb}_2^{(b)}$, except we add the following halting condition:
 - **Generic group oracle queries:** If BGGM.Encode , BGGM.Add , or BGGM.Pair returns “collision” because the sampled handle $\langle h \rangle$ has previously been sampled (i.e., $\mathcal{T}(\langle h \rangle)$ is already defined), halt with output 0.
- **Hybrid $\text{Hyb}_4^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_3^{(b)}$ (including its handle collision-abort rule) except that it augments the challenger so that, in addition to maintaining handles that refer to group elements, it also keeps track of two kinds of formal polynomials to which the handles map. Specifically, handles given to the adversary map to polynomials over a set of formal variables $\hat{\mathcal{V}}$, while the challenger simultaneously maintains polynomials over a set of formal variables $\tilde{\mathcal{V}}$ that it will use to evaluate zero-test queries. We refer to the former as the *adversary’s* formal polynomials and the latter as the *challenger’s* formal polynomials. To distinguish the representations used below, a plain symbol x denotes an actual exponent in $\mathbb{Z}_p$, $[x]_v$ denotes the corresponding concrete group element in group $\mathbb{G}_v$, and $\langle x \rangle$ denotes an opaque handle for that group element in the generic-group game. A hatted polynomial $\hat{p}$ records, using only the components exposed to the adversary, the formal algebraic expression associated with a handle, while a tilded polynomial $\tilde{p}$ expands the same handle in terms of the challenger’s underlying sampled values.

The set of variables $\hat{\mathcal{V}}$ over which the adversary’s formal polynomials are defined captures the components the construction returns during setup, key generation, and encryption. In detail, $\hat{\mathcal{V}}$ is specified as follows:

$$\begin{aligned}\hat{\mathcal{V}}_1 &= \{\hat{c}_2^*\} \cup \{\hat{w}_i, \tau \hat{w}_i, \hat{c}_{3,i}^*\}_{i \in [K]} \cup \{\hat{y}_i, \hat{c}_{4,i}^*\}_{i \in [L]}, \\ \hat{\mathcal{V}}_2 &= \{\hat{\tau}^i\}_{i \in [B]} \cup \{\hat{\delta}_\xi\}_{\xi \in [q_{\text{KG}}]} \cup \{\hat{\beta}_{i,\xi}\}_{(i,\xi) \in [L] \times [q_{\text{KG}}]}, \\ \hat{\mathcal{V}}_{\text{T}} &= \{\hat{\alpha}, \hat{c}_1^*\}, \\ \hat{\mathcal{V}} &= \hat{\mathcal{V}}_1 \cup \hat{\mathcal{V}}_2 \cup \hat{\mathcal{V}}_{\text{T}}.\end{aligned}$$

Thus, $\hat{\mathcal{V}}$ contains variables only for handles supplied directly by the challenger. Handles produced by pairing are instead represented by degree-two monomials in these variables. Consequently, a formal polynomial associated with a handle in $\mathbb{G}_{\text{T}}$ lies in the linear span of $\{1\} \cup \hat{\mathcal{V}}_1 \cup \hat{\mathcal{V}}_2 \cup \hat{\mathcal{V}}_{\text{T}} \cup \{\hat{v}_1 \hat{v}_2 : (\hat{v}_1, \hat{v}_2) \in \hat{\mathcal{V}}_1 \times \hat{\mathcal{V}}_2\}$.

The set of variables $\tilde{\mathcal{V}}$ over which the challenger’s formal polynomials are defined captures the group elements sampled when running setup, key generation, and encryption. In detail, $\tilde{\mathcal{V}}$ is specified as follows:

$$\tilde{\mathcal{V}} = \{\tilde{\alpha}, \tilde{\tau}, \tilde{s}\} \cup \{\tilde{w}_i\}_{i \in [K]} \cup \{\tilde{y}_i, \tilde{r}_i\}_{i \in [L]} \cup \{\tilde{\beta}_{i,\xi}\}_{(i,\xi) \in [L] \times [q_{\text{KG}}]}.$$

At any time t in the hybrid, define an assignment $\phi_t: \tilde{\mathcal{V}} \rightarrow \mathbb{Z}_p$ that maps each challenger variable $\tilde{v} \in \tilde{\mathcal{V}}$ to the corresponding value $v \in \mathbb{Z}_p$ if it has been sampled by time t and to 0 otherwise. We let $\text{Eval}_{\phi_t}: \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p$ denote the standard polynomial-evaluation homomorphism induced by ϕ_t . Thus, in particular, the variables $\tilde{r}_i$ and $\tilde{\beta}_{i,\xi}$ evaluate to 0 until the corresponding values have been sampled.

With adversary and challenger formal polynomials defined, the hybrid proceeds as follows:

- **Setup phase:** The challenger performs the setup procedure with the following additional steps:
 - * When the challenger initializes the table $\mathcal{T}$, it additionally initializes tables $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$.
 - * With handles $\langle w_i \rangle$ and $\langle \tau w_i \rangle$ for each $i \in [K]$, it adds the entries $\langle w_i \rangle \mapsto \hat{w}_i$ and $\langle \tau w_i \rangle \mapsto \hat{\tau} \hat{w}_i$ to $\hat{\mathcal{T}}$ and adds the entries $\langle w_i \rangle \mapsto \tilde{w}_i$ and $\langle \tau w_i \rangle \mapsto \tilde{\tau} \tilde{w}_i$ to $\tilde{\mathcal{T}}$. With the handle $\langle \tau^0 \rangle$, it adds the entry $\langle \tau^0 \rangle \mapsto 1$ to $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$. Next, with handles $\langle \tau^j \rangle$ for each $j \in [B]$, it adds the entry $\langle \tau^j \rangle \mapsto \hat{\tau}^j$ to $\hat{\mathcal{T}}$ and adds the entry $\langle \tau^j \rangle \mapsto \tilde{\tau}^j$ to $\tilde{\mathcal{T}}$. Let $\tilde{\mathbf{w}} = (\tilde{w}_1, \dots, \tilde{w}_K)$ and $\tilde{\tau} = (1, \tilde{\tau}, \tilde{\tau}^2, \dots, \tilde{\tau}^B)$.
 - * Finally, when it produces handles $\langle \alpha \rangle, \langle \gamma_1 \rangle, \dots, \langle \gamma_L \rangle$, it adds the entries $\langle \alpha \rangle \mapsto \hat{\alpha}$ and $\langle \gamma_i \rangle \mapsto \hat{\gamma}_i$ for each $i \in [L]$ to $\hat{\mathcal{T}}$, and it also adds the entries $\langle \alpha \rangle \mapsto \tilde{\alpha}$ and $\langle \gamma_i \rangle \mapsto \tilde{\gamma}_i$ for each $i \in [L]$ to $\tilde{\mathcal{T}}$.
- **Pre-challenge key-generation phase:** For a ξ^{th} key-generation query on set I_ξ , the challenger generates an identity-batch secret key with the following additional steps:
 - * When the challenger produces the handle $\langle \beta_{i,\xi} \rangle$ for each $i \in S_\xi$, it adds the entry $\langle \beta_{i,\xi} \rangle \mapsto \hat{\beta}_{i,\xi}$ to $\hat{\mathcal{T}}$ and also adds the entry $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ to $\tilde{\mathcal{T}}$.
 - * When it computes $\delta_\xi = \alpha + \mathbf{y}_\xi^T \mathbf{w} \cdot \mathbf{f}_\xi^T \tau + \sum_{i \in S_\xi} \gamma_i \beta_{i,\xi}$ to produce the handle $\langle \delta_\xi \rangle$, it additionally adds the entry $\langle \delta_\xi \rangle \mapsto \hat{\delta}_\xi$ to $\hat{\mathcal{T}}$ and adds the entry $\langle \delta_\xi \rangle \mapsto \tilde{\alpha} + (\sum_{i \in [K]} \gamma_{i,\xi} \tilde{w}_i) \cdot F_\xi(\tilde{\tau}) + \sum_{i \in S_\xi} \tilde{\gamma}_i \tilde{\beta}_{i,\xi}$ to $\tilde{\mathcal{T}}$, where $F_\xi(\tilde{\tau}) := \prod_{\text{id} \in I_\xi} (\tilde{\tau} - \text{id})$ is the polynomial associated with $\mathbf{f}_\xi$.

When $\mathcal{A}$ makes generic group oracle queries, the challenger performs the following additional steps:

- * **BGGM.Encode(x, v):** In addition to adding the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger adds the entry $\langle h \rangle \mapsto x$ to both $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ where x denotes the constant polynomial.
- * **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** When it adds the entry $\langle h \rangle \mapsto (x_1 + x_2, v_1)$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger additionally adds the entries $\langle h \rangle \mapsto \hat{\mathcal{T}}(\langle h_1 \rangle) + \hat{\mathcal{T}}(\langle h_2 \rangle)$ to $\hat{\mathcal{T}}$ and $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$, where both sums denote formal polynomial addition.
- * **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** When it adds the entry $\langle h \rangle \mapsto (x_1 x_2, \mathbf{T})$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger additionally adds the entries $\langle h \rangle \mapsto \hat{\mathcal{T}}(\langle h_1 \rangle) \cdot \hat{\mathcal{T}}(\langle h_2 \rangle)$ to $\hat{\mathcal{T}}$ and $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$, where both products denote multiplication in the formal polynomial ring.
- * **BGGM.ZeroTest($\langle h \rangle$):** If $\langle h \rangle$ is present in $\mathcal{T}$, then it returns “zero” if

$$\text{Eval}_{\phi_t}(\tilde{h}) = 0$$

for the challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ at the time t of the query, and returns “non-zero” otherwise.

- **Challenge phase:** This phase includes the following additional steps:
 - * When the challenger produces the handle $\langle c_1^* \rangle$, it also adds the entries $\langle c_1^* \rangle \mapsto \hat{c}_1^*$ to $\hat{\mathcal{T}}$ and $\langle c_1^* \rangle \mapsto \tilde{s} \tilde{\alpha} + \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ to $\tilde{\mathcal{T}}$ where $\langle m_b^* \rangle$ is the handle for the challenge message.
 - * With the handle $\langle c_2^* \rangle$, it adds the entry $\langle c_2^* \rangle \mapsto \hat{c}_2^*$ to $\hat{\mathcal{T}}$ and the entry $\langle c_2^* \rangle \mapsto \tilde{s}$ to $\tilde{\mathcal{T}}$.
 - * For each $i \in [K]$, with $\langle c_{3,i}^* \rangle$, it adds the entry $\langle c_{3,i}^* \rangle \mapsto \hat{c}_{3,i}^*$ to $\hat{\mathcal{T}}$ and $\langle c_{3,i}^* \rangle \mapsto \tilde{s}(\tilde{\tau} - \text{id}^*) \tilde{w}_i$ to $\tilde{\mathcal{T}}$.
 - * Finally, for the handles $\langle c_{4,i}^* \rangle$ for each $i \in [L]$, it adds the entry $\langle c_{4,i}^* \rangle \mapsto \hat{c}_{4,i}^*$ to $\hat{\mathcal{T}}$. It additionally adds to $\tilde{\mathcal{T}}$ the entry $\langle c_{4,i}^* \rangle \mapsto \tilde{s} \tilde{\gamma}_i$ if $i \in S^*$ and $\langle c_{4,i}^* \rangle \mapsto \tilde{r}_i$ otherwise.
- **Post-challenge key-generation phase:** This phase is the same as the pre-challenge key-generation phase. In particular, at the time t of each zero-test query, the challenger uses the current assignment ϕ_t , which now includes the values sampled during the challenge phase, and returns “zero” if $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ (where $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ is the challenger formal polynomial associated with the queried handle $\langle h \rangle$) and “non-zero” otherwise.

- **Hybrid $\text{Hyb}_5^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_4^{(b)}$, except for the following change:
 - **Zero-test oracle queries:** When the adversary $\mathcal{A}$ queries a handle $\langle h \rangle$ to the zero-test oracle, the challenger no longer evaluates the associated challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ under the current assignment ϕ_t . Instead, it **returns “zero”** if $\tilde{\mathcal{T}}(\langle h \rangle) \equiv 0$ in $\mathbb{Z}_p[\tilde{\mathcal{V}}]$ and “non-zero” otherwise. That is, this hybrid tests whether $\tilde{\mathcal{T}}(\langle h \rangle)$ is identically zero as a formal polynomial, while previous hybrids tested whether $\text{Eval}_{\phi_t}(\tilde{\mathcal{T}}(\langle h \rangle)) = 0$.
- **Hybrid $\text{Hyb}_{6,0,\text{int}}^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_5^{(b)}$, except for the following change:
 - **Challenge phase:** For each handle $\langle c_{4,i}^* \rangle$ for $i \in [L]$, the challenger always **adds to $\tilde{\mathcal{T}}$ the entry $\langle c_{4,i}^* \rangle \mapsto \tilde{r}_i$** .² Previously, it sampled a set S^* and added the entry $\langle c_{4,i}^* \rangle \mapsto \tilde{s}_i$ to $\tilde{\mathcal{T}}$ if $i \in S^*$ and $\langle c_{4,i}^* \rangle \mapsto \tilde{r}_i$ to $\tilde{\mathcal{T}}$ otherwise.
 - **Post-challenge query phase:** At the start of the post-challenge query phase, the challenger does the following additional steps:
 - * The challenger samples a set $S^* \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$ and defines $\text{Sub}_{S^*} : \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ to be the ring homomorphism induced by

$$\tilde{r}_i \mapsto \tilde{s}_i \quad \text{for } i \in S^* \quad \text{and} \quad \tilde{v} \mapsto \tilde{v} \quad \text{otherwise.}$$

For each handle $\langle h \rangle$ in $\tilde{\mathcal{T}}$, let $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ be the challenger’s formal polynomial indexed by $\langle h \rangle$ and define $\tilde{h}' = \text{Sub}_{S^*}(\tilde{h})$ (i.e., the formal challenger polynomial $\tilde{h}'$ is $\tilde{h}$ where the $\tilde{r}_i$ components are substituted with $\tilde{s}_i$ for $i \in S^*$). Replace the entry $\langle h \rangle \mapsto \tilde{h}$ in $\tilde{\mathcal{T}}$ with $\langle h \rangle \mapsto \tilde{h}'$.
- **Hybrid $\text{Hyb}_{6,\chi}^{(b)}(\lambda)$ for $\chi \in [q_{\text{ZT}}]$:** This hybrid is equivalent to $\text{Hyb}_{6,\chi-1,\text{int}}^{(b)}$, except for the following change:
 - **Post-challenge zero-test oracle queries:** On the χ^{th} post-challenge query to the zero-test oracle, the challenger does the following based on the adversary’s formal polynomial $\hat{h} = \hat{\mathcal{T}}(\langle h \rangle)$ associated with the handle $\langle h \rangle$ queried by the adversary:
 - * **Message-dependent component:** If $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$ (i.e., there is a $\hat{c}_1^*$ term in the adversary’s formal polynomial associated with $\langle h \rangle$), **then the oracle returns “non-zero”** (regardless of whether it is identical to the zero polynomial or not).
 - * **Challenge-set-dependent component:** If $\text{coeff}_{\hat{c}_{4,i}^* \hat{v}}(\hat{h}) \neq 0$ for some $i \in [L]$ and $\hat{v} \in \{1\} \cup \hat{\mathcal{V}}_2$, **then the oracle returns “non-zero”** (regardless of whether it is identical to the zero polynomial or not).
 - * If neither of the cases above occur, and for the remaining queries to the zero-test oracle, the challenger proceeds as in hybrid $\text{Hyb}_{6,\chi-1,\text{int}}^{(b)}$.

Note that because the hybrid is defined recursively, all post-challenge zero-test oracle queries before the χ^{th} query have already also been changed to include the above behavior.
- **Hybrid $\text{Hyb}_{6,\chi,\text{int}}^{(b)}(\lambda)$ for $\chi \in [q_{\text{ZT}}]$:** This hybrid is equivalent to $\text{Hyb}_{6,\chi}^{(b)}$, except for the following change:
 - **Post-challenge zero-test oracle queries:** After the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query, the challenger **no longer samples S^* and performs the Sub_{S^*} substitution**. Instead, after the χ^{th} post-challenge query to the zero-test oracle, the challenger performs this substitution, explicitly doing the following:
 - * Sample a set $S^* \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. For each handle $\langle h \rangle$ in $\tilde{\mathcal{T}}$, let $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ be the challenger’s formal polynomial indexed by $\langle h \rangle$ and let $\tilde{h}' = \text{Sub}_{S^*}(\tilde{h})$ (i.e., $\tilde{h}'$ is the polynomial $\tilde{h}$ where the $\tilde{r}_i$ components are substituted with $\tilde{s}_i$ for $i \in S^*$). Replace the entry $\langle h \rangle \mapsto \tilde{h}$ in $\tilde{\mathcal{T}}$ with the entry $\langle h \rangle \mapsto \tilde{h}'$.

In the previous hybrid, the step was performed after the first $\chi - 1$ post-challenge queries to the zero-test oracle, instead of the χ^{th} post-challenge query.

²Note that the behavior of the hybrid on actual exponents is irrelevant to the security analysis, as nothing computed from these exponents is given to the adversary in this hybrid or later. Thus, for simplicity, we elide this detail in the specification of the hybrid.

Lemma 3.11. For $K = \text{poly}_2(\lambda)$, $P = \omega(K \cdot \log \lambda) \geq \lambda$, and $2^{P(\lambda)} < p = 2^{\Theta(P(\lambda))}$, there exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,

$$|\Pr[\text{Hyb}_0^{(b)}(\lambda)] - \Pr[\text{Hyb}_1^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The hybrids are equivalent except for the event that the halting condition in $\text{Hyb}_1^{(b)}$ is triggered. Namely, the experiments are equivalent unless some collection of K or fewer vectors $\mathbf{y}_{\xi_1}, \dots, \mathbf{y}_{\xi_{K'}}$ (i.e., with $K' \leq K$) sampled during (pre- or post-challenge) key-generation are linearly dependent. Let bad denote this event.

We first consider the probability that a fixed subset $D = \{\xi_1, \dots, \xi_{K'}\} \subseteq [q_{\text{KG}}]$ triggers the halting condition. Let $\mathbf{Y}_D = (\mathbf{y}_{\xi_1}, \dots, \mathbf{y}_{\xi_{K'}})$. The halting condition is equivalent to the event that $\text{rank}(\mathbf{Y}_D) < K'$. We have that

$$\Pr[\text{rank}(\mathbf{Y}_D) < K'] \leq \sum_{i \in [K']} \frac{1}{p^{K-i+1}} \leq \frac{K'}{p^{K-K'+1}} \leq \frac{K'}{p}.$$

To analyze the full bad event, let $\Delta = \{D \in 2^{[q_{\text{KG}}]} : |D| \leq K\}$ be the set of all possible subsets D of size at most K . By the union bound, we have that

$$\begin{aligned} \Pr[\text{bad}] &\leq \sum_{D \in \Delta} \Pr[\text{rank}(\mathbf{Y}_D) < |D|] \leq \sum_{K' \in [K]} \binom{q_{\text{KG}}}{K'} \cdot \frac{K'}{p} \\ &\leq \sum_{K' \in [K]} q_{\text{KG}}^{K'} \cdot \frac{K'}{p} \leq q_{\text{KG}}^K \cdot \frac{K^2}{p} \\ &= 2^{K \log q_{\text{KG}} + 2 \log K - \log p}. \end{aligned}$$

Finally, recall that $K = \text{poly}_2(\lambda) \leq \lambda^{c_2}$ for all $\lambda \geq d_2$ and $q_{\text{KG}} = \text{poly}_3(\lambda) \leq \lambda^{c_3}$ for all $\lambda \geq d_3$ for some $c_2, c_3, d_2, d_3 \in \mathbb{N}$, and that $P = \omega(K \cdot \log \lambda)$ and $2^{P(\lambda)} < p$. Consider any $c_4 > 0$. Because $P = \omega(K \cdot \log \lambda)$, there exists some $d_4 \in \mathbb{N}$ such that $P(\lambda) \geq c_4 \cdot K \cdot \log \lambda$ for all $\lambda \geq d_4$. Let $c = c_4 - c_3 - 2c_2$ and $d = \max(d_2, d_3, d_4)$. We have that,

$$\begin{aligned} \Pr[\text{bad}] &\leq 2^{c_3 K \log \lambda + 2c_2 \log \lambda - P(\lambda)} && \text{for } \lambda \geq \max(d_2, d_3) \\ &\leq 2^{(c_3 + 2c_2) \cdot K \log \lambda - c_4 \cdot K \log \lambda} && \text{for } \lambda \geq d \\ &= \lambda^{-K \cdot c} && \text{for } \lambda \geq d \\ &\leq \lambda^{-c} && \text{for } \lambda \geq d, \text{ if } c \geq 0. \end{aligned}$$

Note that since $c_4 > 0$ was arbitrary, c can take any positive value. That is, for every $c > 0$, it holds that $\Pr[\text{bad}] \leq \lambda^{-c}$ for all sufficiently large λ , and so there exists some negligible function $\text{negl}(\cdot)$ such that

$$|\Pr[\text{Hyb}_0^{(b)}(\lambda)] - \Pr[\text{Hyb}_1^{(b)}(\lambda)]| \leq \Pr[\text{bad}] \leq \text{negl}(\lambda). \quad \square$$

Lemma 3.12. For $L = \text{poly}_1(\lambda)$ and $K = \text{poly}_2(\lambda)$, $L_{\text{sk}} = \omega(\log \lambda)$ and $K \cdot L_{\text{sk}} \geq \kappa \cdot L$ for some constant $\kappa \geq 2$, there exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,

$$|\Pr[\text{Hyb}_1^{(b)}(\lambda)] - \Pr[\text{Hyb}_2^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The hybrids are equivalent except for the event that the halting condition in $\text{Hyb}_2^{(b)}$ is triggered. Namely, the experiments are equivalent unless there exists some collection of sets $S_{\xi_1}, \dots, S_{\xi_K} \subseteq [L]$ sampled during (pre- or post-challenge) key-generation for K different keys $\xi_1, \dots, \xi_K$ such that the union $|\bigcup_{i \in [K]} S_{\xi_i}| \leq L/\kappa$, or equivalently, unless there exists some set $T \subseteq [L]$ with $|T| = L/\kappa$ such that $S_{\xi_i} \subseteq T$ for all $i \in [K]$. Let bad denote this event.

We first consider the probability that a given set $S_{\xi_j} \stackrel{\mathcal{R}}{\leftarrow} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$ is contained within a fixed set T of size L/κ . We have that

$$\Pr[S_{\xi_j} \subseteq T] \leq \binom{L/\kappa}{L_{\text{sk}}} \cdot \left(\frac{L}{L_{\text{sk}}}\right)^{-1} \leq \left(\frac{L/\kappa}{L}\right)^{L_{\text{sk}}} = \kappa^{-L_{\text{sk}}}.$$

Now, because each set $S_{\xi_1}, \dots, S_{\xi_K}$ is sampled independently of each other (and any queries the adversary does, as each key-generation query samples a fresh set), we have that

$$\Pr[\forall i \in [K] : S_{\xi_i} \subseteq T] = \Pr[S_{\xi_j} \subseteq T]^K \leq \kappa^{-K \cdot L_{\text{sk}}}.$$

To analyze the full bad event, observe that there are $\binom{q_{\text{KG}}}{K}$ possible collections of K sets and $\binom{L}{L/\kappa}$ possible target sets T . By applying the union bound and using the fact that $\kappa \geq 2$, we have that,

$$\begin{aligned} \Pr[\text{bad}] &\leq \binom{q_{\text{KG}}}{K} \cdot \binom{L}{L/\kappa} \cdot \Pr[\forall i \in [K] : S_{\xi_i} \subseteq T] \\ &\leq \left(\frac{e \cdot q_{\text{KG}}}{K}\right)^K \cdot \kappa^L \cdot \kappa^{-K \cdot L_{\text{sk}}} \\ &\leq \kappa^{K \cdot \log(e \cdot q_{\text{KG}}/K) / \log \kappa + L - K \cdot L_{\text{sk}}}. \end{aligned}$$

Finally, recall that $L = \text{poly}_1(\lambda) \leq \lambda^{c_1}$ for all $\lambda \geq d_1$, $K = \text{poly}_2(\lambda) \leq \lambda^{c_2}$ for all $\lambda \geq d_2$, and $q_{\text{KG}} = \text{poly}_3(\lambda) \leq \lambda^{c_3}$ for all $\lambda \geq d_3$ for some $c_1, c_2, c_3, d_1, d_2, d_3 \in \mathbb{N}$, and that $K \cdot L_{\text{sk}} \geq \kappa \cdot L$. Consider any $c_4 > 0$. Because $L_{\text{sk}} = \omega(\log \lambda)$, there exists some $d_4 \in \mathbb{N}$ such that $L_{\text{sk}} \geq c_4 \log \lambda$ for all $\lambda \geq d_4$. Let $c = (1 - 1/\kappa) \log \kappa \cdot c_4 - c_3 - 1$ and $d = \max(d_3, d_4, 3)$. We have that,

$$\begin{aligned} \Pr[\text{bad}] &\leq \kappa^{K \cdot \log(e \cdot \lambda^{c_3}) / \log \kappa + K \cdot L_{\text{sk}} / \kappa - K \cdot L_{\text{sk}}} && \text{for } \lambda \geq d_3 \\ &\leq \kappa^{K \cdot ((c_3+1) \log \lambda / \log \kappa - (1-1/\kappa) \cdot c_4 \log \lambda)} && \text{for } \lambda \geq d \\ &= \lambda^{-K \cdot c} && \text{for } \lambda \geq d \\ &\leq \lambda^{-c} && \text{for } \lambda \geq d, \text{ if } c \geq 0. \end{aligned}$$

Note that since $c_4 > 0$ was arbitrary, c can take any positive value. That is, for every $c > 0$, it holds that $\Pr[\text{bad}] \leq \lambda^{-c}$ for all sufficiently large λ , and so there exists some negligible function $\text{negl}(\cdot)$ such that

$$|\Pr[\text{Hyb}_1^{(b)}(\lambda)] - \Pr[\text{Hyb}_2^{(b)}(\lambda)]| \leq \Pr[\text{bad}] \leq \text{negl}(\lambda). \quad \square$$

Lemma 3.13. For $L = \text{poly}_1(\lambda)$ and $K = \text{poly}_2(\lambda)$, there exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,

$$|\Pr[\text{Hyb}_2^{(b)}(\lambda)] - \Pr[\text{Hyb}_3^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The two hybrids are identical except for the event that the halting condition in $\text{Hyb}_3^{(b)}$ is triggered (i.e., one of the handles sampled during the experiment is equal to a previously sampled handle). Let bad denote this event. Since each handle is sampled uniformly from $\mathcal{H} = \{0, 1\}^\lambda$, a union bound over pairs of sampled handles over the course of the hybrid (both sampled during adversarial queries and by the challenger) gives us

$$\Pr[\text{bad}] \leq 2^{-\lambda} \cdot \left(\underbrace{2K + L + B + 2}_{\text{Setup phase}} + \underbrace{q_{\text{KG}} \cdot (L_{\text{sk}} + 1)}_{\text{Key generation queries}} + \underbrace{q_{\text{BGGM}}}_{\text{Generic group queries}} + \underbrace{K + L + 3}_{\text{Challenge phase}} \right)^2.$$

Finally, recall that $L = \text{poly}_1(\lambda) \leq \lambda^{c_1}$ for all $\lambda \geq d_1$, $K = \text{poly}_2(\lambda) \leq \lambda^{c_2}$ for all $\lambda \geq d_2$, $q_{\text{KG}} = \text{poly}_3(\lambda) \leq \lambda^{c_3}$ for all $\lambda \geq d_3$, $q_{\text{BGGM}} = \text{poly}_4(\lambda) \leq \lambda^{c_4}$ for all $\lambda \geq d_4$, and $B = \text{poly}_5(\lambda) \leq \lambda^{c_5}$ for all $\lambda \geq d_5$ (because the adversary declares 1^B in unary) for some $c_1, c_2, c_3, c_4, c_5, d_1, d_2, d_3, d_4, d_5 \in \mathbb{N}$. Furthermore, $L_{\text{sk}} \leq L$. Let $c = \max(c_1, c_2, c_3, c_4, c_5)$ and $d = \max(d_1, d_2, d_3, d_4, d_5)$. We have that

$$\begin{aligned} \Pr[\text{bad}] &\leq 2^{-\lambda} \cdot \left(2\lambda^{c_2} + \lambda^{c_1} + \lambda^{c_5} + 2 + \lambda^{c_3}(\lambda^{c_1} + 1) + \lambda^{c_4} + \lambda^{c_2} + \lambda^{c_1} + 3 \right)^2 && \text{for } \lambda \geq d \\ &\leq 2^{-\lambda} \cdot \left(\lambda^{2c} + 8\lambda^c + 5 \right)^2 && \text{for } \lambda \geq d \\ &\leq 2^{-\lambda} \cdot (16\lambda^{2c})^2 && \text{for } \lambda \geq d \\ &= 2^{8-\lambda+4c \log \lambda} && \text{for } \lambda \geq d \\ &\leq 2^{-\frac{1}{2}\lambda} && \text{for } \lambda \geq \max(d, 256c^2, 32). \end{aligned}$$

It follows that there exists some negligible function $\text{negl}(\cdot)$ such that

$$|\Pr[\text{Hyb}_2^{(b)}(\lambda)] - \Pr[\text{Hyb}_3^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

□

Lemma 3.14. For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_3^{(b)}(\lambda)] = \Pr[\text{Hyb}_4^{(b)}(\lambda)]$.

Proof. The two hybrids are identical except that, in hybrid $\text{Hyb}_4^{(b)}$, the challenger additionally maintains adversary and challenger formal polynomial tables $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ respectively, and it answers a zero-test query on a handle $\langle h \rangle$ at time t by testing whether $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ for the challenger's formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$. In hybrid $\text{Hyb}_3^{(b)}$, the challenger instead tested whether $x = 0$ for the exponent x where $\mathcal{T}(\langle h \rangle) = (x, v)$. Because neither table is given to the adversary $\mathcal{A}$ and neither is used to answer any other oracle query, we need only argue that these two ways of answering a zero-test query always agree. We do so by showing that the challenger's formal polynomial $\tilde{h}$ associated with a handle $\langle h \rangle$ always evaluates to the exponent x held in $\mathcal{T}(\langle h \rangle)$.

Before continuing with the analysis, we first recall the entries the challenger adds to $\tilde{\mathcal{T}}$ for the handles it produces over the course of the experiment:

$$\begin{aligned} \mathbb{G}_1 \left\{ \begin{array}{ll} \langle c_2^* \rangle \mapsto \tilde{s} & \\ \langle w_i \rangle \mapsto \tilde{w}_i & \text{for } i \in [K] \\ \langle \tau w_i \rangle \mapsto \tilde{\tau} \tilde{w}_i & \text{for } i \in [K] \\ \langle \gamma_i \rangle \mapsto \tilde{\gamma}_i & \text{for } i \in [L] \end{array} \right. & \begin{array}{ll} \langle c_{3,i}^* \rangle \mapsto \tilde{s}(\tilde{\tau} - \text{id}^*) \tilde{w}_i & \text{for } i \in [K] \\ \langle c_{4,i}^* \rangle \mapsto \begin{cases} \tilde{s} \tilde{\gamma}_i & i \in S^* \\ \tilde{r}_i & \text{otherwise} \end{cases} & \text{for } i \in [L], \end{array} \\ \mathbb{G}_2 \left\{ \begin{array}{ll} \langle \tau^0 \rangle \mapsto 1 & \\ \langle \tau^j \rangle \mapsto \tilde{\tau}^j & \text{for } j \in [B] \\ \langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi} & \text{for } \xi \in [q_{KG}], i \in S_\xi \end{array} \right. & \langle \delta_\xi \rangle \mapsto \tilde{\alpha} + \left(\sum_{i \in [K]} y_{i,\xi} \tilde{w}_i \right) \cdot F_\xi(\tilde{\tau}) + \sum_{i \in S_\xi} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} \quad \text{for } \xi \in [q_{KG}], \\ \mathbb{G}_T \left\{ \begin{array}{ll} \langle \alpha \rangle \mapsto \tilde{\alpha} & \\ \langle c_1^* \rangle \mapsto \tilde{s} \tilde{\alpha} + \tilde{m}_b^* & \end{array} \right. & \end{aligned} \quad (3.1)$$

where $\tilde{m}_b^* = \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ is the challenger's formal polynomial associated with the challenge message handle $\langle m_b^* \rangle$. We also recall that ϕ_t maps each challenger variable to its corresponding sampled value (i.e., $\tilde{\alpha} \mapsto \alpha$, $\tilde{\tau} \mapsto \tau$, $\tilde{s} \mapsto s$, $\tilde{w}_i \mapsto w_i$, $\tilde{\gamma}_i \mapsto \gamma_i$, $\tilde{\beta}_{i,\xi} \mapsto \beta_{i,\xi}$, and $\tilde{r}_i \mapsto c_{4,i}^*$ for $i \notin S^*$), provided the value has been sampled by time t (otherwise, it is mapped to 0).

We now argue that $\text{Eval}_{\phi_t}(\tilde{h}) = x$ for every handle $\langle h \rangle$ with $\mathcal{T}(\langle h \rangle) = (x, v)$ and every time $t \geq t_h$, where t_h denotes the time at which $\langle h \rangle$ is created. For this, it suffices to show that $\text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$, as $\text{Eval}_{\phi_t}(\tilde{h}) = \text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$ for all $t \geq t_h$ because once an actual exponent is sampled, it is never reassigned. Our analysis breaks down into a case covering handles mapping to values directly sampled by the challenger (i.e., handles output by the challenger during setup, key-generation queries, and the challenge phase, excluding the message-dependent challenge-ciphertext component) which we call *directly generated handles*, and a case covering handles mapping to values computed from values stored in other handles (i.e., handles produced by generic group oracle queries as well as the message-dependent challenge-ciphertext component) which we call *derived handles*.

- **Case I: $\langle h \rangle$ is directly generated:** For this case, we can see from Eq. (3.1) that, by construction, each handle produced by the challenger during these phases maps directly to the polynomial that, when substituted with the appropriate variables (as is done when executing $\text{Eval}_{\phi_{t_h}}(\tilde{h})$), exactly computes what the challenger computes to produce these exponents. We accordingly only need to argue that the appropriate variables have been sampled before time t_h , as they are otherwise substituted with 0.

For handles $\langle \alpha \rangle$, $\langle \gamma_i \rangle$, $\langle w_i \rangle$, $\langle \beta_{i,\xi} \rangle$, and $\langle c_2^* \rangle$, the challenger adds the sampled exponents α , γ_i , w_i , $\beta_{i,\xi}$, and s , to $\mathcal{T}$ directly before adding the challenger formal polynomials $\tilde{\alpha}$, $\tilde{\gamma}_i$, $\tilde{w}_i$, $\tilde{\beta}_{i,\xi}$, and $\tilde{s}$ to $\tilde{\mathcal{T}}$. Similarly, for handles $\langle \tau w_i \rangle$ and $\langle \tau^j \rangle$ produced during setup and a handle $\langle c_{3,i}^* \rangle$ produced during the challenge phase, the challenger encodes τw_i , τ^j , and $s(\tau - \text{id}^*) w_i$, whose values τ , w_i , and s were all sampled prior in setup or in the challenge phase.

For a handle $\langle c_{4,i}^* \rangle$, recall that the challenger computes the exponent $c_{4,i}^* = sy_i$ if $i \in S^*$ and samples $c_{4,i}^* \xleftarrow{R} \mathbb{Z}_p$ otherwise, and that this occurs in the challenge phase. In the former case, the entry is $\tilde{sy}_i$ and both s and y_i have been sampled by this point (in the challenge phase and setup phase respectively), while in the latter case, the entry is $\tilde{r}_i$ which is added to $\tilde{\mathcal{T}}$ directly after sampling $c_{4,i}^*$.

Finally, for a handle $\langle \delta_\xi \rangle$ which is sampled during a key-generation query, the challenger computes the exponent $\delta_\xi = \alpha + \mathbf{y}_\xi^\top \mathbf{w} \cdot \mathbf{f}_\xi^\top \boldsymbol{\tau} + \sum_{i \in S_\xi} y_i \beta_{i,\xi}$, and the values $\alpha, \tau, w_1, \dots, w_K$, and $y_i, \beta_{i,\xi}$ for $i \in S_\xi$ were all sampled prior (in setup or earlier in the key-generation query in question).

- **Case II: $\langle h \rangle$ is derived:** We use induction on the order in which the handles were sampled.

The base case is the first handle created over the course of the experiment, which is a handle $\langle w_i \rangle$ that the challenger produces during setup. This is directly covered by Case I.

For the inductive step, we first consider each of the three generic group oracles separately. For the add and pair oracles, we write $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$, and we note that the handles $\langle h_1 \rangle$ and $\langle h_2 \rangle$ must be sampled before $\langle h \rangle$ for the oracle to not abort with output “invalid”. Thus, by the induction hypothesis, we have that $\text{Eval}_{\phi_{t_h}}(\tilde{h}_1) = x_1$ and $\text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_2$ at the time t_h .

- **BGGM.Encode(x, v):** The challenger adds the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, thus $\text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$.
- **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** The challenger adds the entry $\langle h \rangle \mapsto (x_1 + x_2, v_1)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto \tilde{h}_1 + \tilde{h}_2$ to $\tilde{\mathcal{T}}$. Notably, every variable in $\tilde{h}_1 + \tilde{h}_2$ appears in $\tilde{h}_1$ or in $\tilde{h}_2$, and has thus already been sampled by time t_{h_1} or t_{h_2} where $t_{h_1}, t_{h_2} < t_h$. Furthermore, because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{h}_1 + \tilde{h}_2) = \text{Eval}_{\phi_{t_h}}(\tilde{h}_1) + \text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_1 + x_2.$$

- **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** The challenger adds the entry $\langle h \rangle \mapsto (x_1 x_2, T)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto \tilde{h}_1 \cdot \tilde{h}_2$ to $\tilde{\mathcal{T}}$. Notably, every variable in $\tilde{h}_1 \cdot \tilde{h}_2$ appears in $\tilde{h}_1$ or in $\tilde{h}_2$, and has thus already been sampled by time t_{h_1} or t_{h_2} where $t_{h_1}, t_{h_2} < t_h$. Furthermore, because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{h}_1 \cdot \tilde{h}_2) = \text{Eval}_{\phi_{t_h}}(\tilde{h}_1) \cdot \text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_1 x_2.$$

Finally, we analyze the handle $\langle c_1^* \rangle$, which the challenger produces during the challenge phase by executing $\text{BGGM.Encode}(s\alpha + m_b^*, T)$ to add the entry $\langle c_1^* \rangle \mapsto (s\alpha + m_b^*, T)$ to $\mathcal{T}$, where $\mathcal{T}(\langle m_b^* \rangle) = (m_b^*, T)$ is the entry for the challenge message handle. The challenger also adds $\langle c_1^* \rangle \mapsto \tilde{s}\alpha + \tilde{m}_b^*$ to $\tilde{\mathcal{T}}$. As above, the values s and α were both sampled prior in the challenge phase or in setup. Furthermore, the challenge message handle $\langle m_b^* \rangle$ must have been sampled before the adversary $\mathcal{A}$ submits it and thus before $\langle c_1^* \rangle$ is sampled. By the induction hypothesis $\text{Eval}_{\phi_{t_h}}(\tilde{m}_b^*) = m_b^*$, and because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{s}\alpha + \tilde{m}_b^*) = s\alpha + m_b^*.$$

As we have shown above, the challenger’s formal polynomial $\tilde{h}$ associated with a handle $\langle h \rangle$ always evaluates to the exponent x held in $\mathcal{T}(\langle h \rangle) = (x, v)$, and the lemma follows. Before continuing, we remark that we can also

consider a mapping from the adversary's formal polynomials to challenger formal polynomials given by:

$$\begin{aligned}
\mathbb{G}_1 \left\{ \begin{array}{ll} \hat{c}_2^* \mapsto \tilde{s} & \\ \hat{w}_i \mapsto \tilde{w}_i & \text{for } i \in [K] \\ \widehat{\tau w}_i \mapsto \tilde{\tau} \tilde{w}_i & \text{for } i \in [K] \\ \hat{y}_i \mapsto \tilde{y}_i & \text{for } i \in [L] \end{array} \right. & \quad \hat{c}_{3,i}^* \mapsto \tilde{s}(\tilde{\tau} - \text{id}^*) \tilde{w}_i & \text{for } i \in [K] \\
& \quad \hat{c}_{4,i}^* \mapsto \begin{cases} \tilde{s} \tilde{y}_i & i \in S^* \\ \tilde{r}_i & \text{otherwise} \end{cases} & \text{for } i \in [L], \\
\mathbb{G}_2 \left\{ \begin{array}{ll} \widehat{\tau} \mapsto \tilde{\tau}^j & \text{for } j \in [B] \\ \hat{\beta}_{i,\xi} \mapsto \tilde{\beta}_{i,\xi} & \text{for } (i, \xi) \in [L] \times [q_{\text{KG}}] \end{array} \right. & \quad \hat{\delta}_\xi \mapsto \tilde{\alpha} + \left(\sum_{i \in [K]} y_{i,\xi} \tilde{w}_i \right) \cdot F_\xi(\tilde{\tau}) + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} & \text{for } \xi \in [q_{\text{KG}}], \\
\mathbb{G}_T \left\{ \begin{array}{l} \hat{\alpha} \mapsto \tilde{\alpha} \end{array} \right. & \quad \hat{c}_1^* \mapsto \tilde{s} \tilde{\alpha} + \tilde{m}_b^*.
\end{aligned} \tag{3.2}$$

where $\tilde{m}_b^* = \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ is the challenger's formal polynomial associated with the challenge message handle $\langle m_b^* \rangle$.

We can extend this to a ring homomorphism $\text{MapP}: \mathbb{Z}_p[\hat{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ in the natural way, such that $\tilde{h} = \text{MapP}(\hat{h})$ for every handle $\langle h \rangle$, where $\hat{h} = \hat{\mathcal{T}}(\langle h \rangle)$ and $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ are the adversary's and the challenger's formal polynomials associated with $\langle h \rangle$. This holds by construction for the handles the challenger produces, as can be seen by inspection of Eq. (3.2), and it is preserved by the generic group oracles, as each of them updates $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ with the same ring operation (a constant polynomial for encode oracle queries, and the sum or the product of the entries of the queried handles for add and pair oracle queries), which MapP respects.

Note that because MapP is a ring homomorphism, it acts on the adversary's formal polynomial term by term (i.e., each term of $\hat{h}$ is mapped to the product of the challenger polynomials of the adversary variables it comprises, and $\tilde{h}$ is the sum of the resulting polynomials). We accordingly say that a term of $\hat{h}$ *induces* the monomials over $\tilde{\mathcal{V}}$ that appear in its image, and, when analyzing the subsequent hybrids, we will study $\tilde{h}$ by asking which terms of $\hat{h}$ can induce a given monomial. Notably, several terms of $\hat{h}$ may induce the same monomial, in which case their contributions to $\tilde{h}$ may cancel. $\square$

Lemma 3.15. *There exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,*

$$|\Pr[\text{Hyb}_4^{(b)}(\lambda)] - \Pr[\text{Hyb}_5^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The hybrids are equivalent except for the event where the zero-test oracle is queried on a handle $\langle h \rangle$ associated with challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ where $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ (so $\text{Hyb}_4^{(b)}$ returns “zero”) but $\tilde{h} \neq 0$ (so $\text{Hyb}_5^{(b)}$ returns “non-zero”). Let bad denote this event. For a particular zero-test query on handle $\langle h' \rangle$ at time t' , we first observe that the associated polynomial $\tilde{h}'$ is at most degree $B+4$. This is because the construction gives out elements as follows:

- In setup, key generation, and encryption, the adversary formal variables in $\hat{\mathcal{V}}_1$ (i.e., the left group $\mathbb{G}_1$) or $\hat{\mathcal{V}}_2$ (i.e., the right group $\mathbb{G}_2$) are sums of products of at most three or $B+1$ challenger variables in $\tilde{\mathcal{V}}$ respectively, so $\mathcal{A}$ cannot pair to produce any polynomial of degree greater than $B+4$.
- Setup and encryption each output a component in the target group $\mathbb{G}_T$; these are, respectively, a single variable in $\tilde{\mathcal{V}}$ and a product of two variables in $\tilde{\mathcal{V}}$ added to a challenge message polynomial produced by pairing other elements (which satisfy the case above).

As hybrids $\text{Hyb}_4^{(b)}$ and $\text{Hyb}_5^{(b)}$ are identical until bad, it suffices to bound $\Pr[\text{bad}]$ in either hybrid [BR06]. In $\text{Hyb}_5^{(b)}$, the zero-test oracle answers by checking whether $\tilde{h} \equiv 0$ and the exponents in $\mathcal{T}$ are never used, so $\mathcal{A}$'s view, and hence each queried polynomial $\tilde{h}'$, is independent of the sampled exponents. We may therefore apply the Schwartz-Zippel lemma (Lemma 2.3) to each zero-test query as though $\tilde{h}'$ were fixed in advance. We have that $\Pr[\text{Eval}_{\phi_{t'}}(\tilde{h}') = 0 \wedge \tilde{h}' \neq 0] \leq (B+4)/p$. The lemma follows by applying the union bound and using the fact that $q_{\text{ZT}} = \text{poly}_3(\lambda)$ and $p \geq 2^{P(\lambda)} \geq 2^\lambda$.

$$\Pr[\text{bad}] \leq q_{\text{ZT}} \cdot \Pr[\text{Eval}_{\phi_{t'}}(\tilde{h}') = 0 \wedge \tilde{h}' \neq 0] \leq \frac{(B+4) \cdot q_{\text{ZT}}}{p} = \text{negl}(\lambda). \quad \square$$

Lemma 3.16. *For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_5^{(b)}(\lambda)] = \Pr[\text{Hyb}_{6,0,\text{int}}^{(b)}(\lambda)]$.*

Proof. All that differs between the two hybrids is when the challenger samples the set S^* and performs the substitution (via the ring homomorphism SubS_{S^*}) on the entries of $\tilde{\mathcal{T}}$. Specifically, this step is performed at the start of the post-challenge query phase in hybrid $\text{Hyb}_{6,0,\text{int}}^{(b)}$, while in hybrid $\text{Hyb}_5^{(b)}$ it is instead performed at the end of the challenge phase. That is, the challenger directly records the substituted entry $\langle c_{4,i}^* \rangle \mapsto \tilde{s}\tilde{y}_i$ for each $i \in S^*$ in hybrid $\text{Hyb}_5^{(b)}$ rather than the entry $\langle c_{4,i}^* \rangle \mapsto \tilde{r}_i$ as is done in hybrid $\text{Hyb}_{6,0,\text{int}}^{(b)}$. We will argue that $\mathcal{A}$'s view is identical in both hybrids.

We first note that, starting in hybrid $\text{Hyb}_5^{(b)}$ (and for all remaining hybrids), the zero-test oracle answers a query on a handle $\langle h \rangle$ by testing whether $\tilde{h} \equiv 0$ (or not checking $\tilde{h}$ at all), and nothing derived from exponents from the table $\mathcal{T}$ is given to $\mathcal{A}$. It follows that the set S^* influences $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$.

Now, observe that $\mathcal{A}$'s view through the end of the challenge phase is identical in both hybrids, as the two hybrids differ only in the entries the challenger records in $\tilde{\mathcal{T}}$ for the handles $\langle c_{4,i}^* \rangle$. This is because $\mathcal{A}$ cannot make any generic group oracle queries between the end of the challenge phase and the substitution point at the start of the post-challenge query phase (notably, no handles are sampled and no zero-test queries are made). In particular, the handles present in $\tilde{\mathcal{T}}$ at the substitution point are exactly those present at the end of the challenge phase.

It remains to argue that the entries of $\tilde{\mathcal{T}}$ coincide in both hybrids from the substitution point onwards. Consider any handle $\langle h \rangle$ present in $\tilde{\mathcal{T}}$ and let $\tilde{h}'$ denote the entry it would have in the absence of any substitution (i.e., the entry obtained by recording $\langle c_{4,i}^* \rangle \mapsto \tilde{r}_i$ for every $i \in [L]$). We argue, by induction on the order in which the handles are created, that the entry $\langle h \rangle \mapsto \text{SubS}_{S^*}(\tilde{h}')$ is inserted in $\tilde{\mathcal{T}}$ in hybrids $\text{Hyb}_5^{(b)}$ and $\text{Hyb}_{6,0,\text{int}}^{(b)}$ and thus the behavior of the hybrids is equivalent.

For the base case, consider a handle $\langle h \rangle$ that is stored in $\tilde{\mathcal{T}}$ at the end of the challenge phase. We begin by analyzing hybrid $\text{Hyb}_5^{(b)}$. If $\langle h \rangle = \langle c_{4,i}^* \rangle$ for some $i \in [L]$, then hybrid $\text{Hyb}_5^{(b)}$ records the entry $\langle h \rangle \mapsto \tilde{s}\tilde{y}_i$ if $i \in S^*$ and $\langle h \rangle \mapsto \tilde{r}_i$ otherwise, which in either case is exactly $\text{SubS}_{S^*}(\tilde{r}_i) = \text{SubS}_{S^*}(\tilde{h}')$ by the definition of SubS_{S^*} . Thus the mapping $\langle h \rangle \mapsto \text{SubS}_{S^*}(\tilde{h}')$ is added to $\tilde{\mathcal{T}}$. Otherwise, $\tilde{h}'$ does not contain any term involving any of the variables $\{\tilde{r}_i\}_{i \in [L]}$. In particular, by inspection of Eq. (3.1), the entries $\langle c_{4,i}^* \rangle \mapsto \tilde{r}_i$ are the only ones the challenger adds that induce such a variable and they are added at the very end of the challenge phase, while every adversary-added entry is added by a generic group oracle query and is either a constant polynomial or a sum or product of entries created earlier. Hence, $\text{SubS}_{S^*}(\tilde{h}') = \tilde{h}'$, and so the entry recorded is again $\langle h \rangle \mapsto \text{SubS}_{S^*}(\tilde{h}')$. Moving on to hybrid $\text{Hyb}_{6,0,\text{int}}^{(b)}$, the base case holds by definition.

For the inductive step, consider a handle $\langle h \rangle$ created after the start of the post-challenge phase and suppose that every handle $\langle h_0 \rangle$ created before it has entry $\text{SubS}_{S^*}(\tilde{h}_0')$. We go over each way $\langle h \rangle$ may have been produced below:

- **Post-challenge key-generation queries:** On the ξ^{th} key-generation query, the challenger adds the entries $\langle \beta_{i,\xi} \rangle \mapsto \beta_{i,\xi}$ for each $i \in S_\xi$ and $\langle \delta_\xi \rangle \mapsto \tilde{\alpha} + (\sum_{i \in [K]} y_{i,\xi} \tilde{w}_i) \cdot F_\xi(\tilde{r}) + \sum_{i \in S_\xi} \tilde{y}_i \beta_{i,\xi}$ to $\tilde{\mathcal{T}}$. Neither entry contains any of the variables $\{\tilde{r}_i\}_{i \in [L]}$ and so both are unaffected by the SubS_{S^*} ring homomorphism.
- **Post-challenge generic group oracle queries:** The analysis of the generic group oracle queries is as follows:
 - **BGGM.Encode(x, v):** The challenger adds the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, which is again unaffected by the SubS_{S^*} ring homomorphism.
 - **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** Recall that the challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that SubS_{S^*} is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{S^*}(\tilde{h}_1') + \text{SubS}_{S^*}(\tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}_1' + \tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}').$$

- **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** The challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that SubS_{S^*} is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{S^*}(\tilde{h}_1') \cdot \text{SubS}_{S^*}(\tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}_1' \cdot \tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}').$$

In each case, the entry $\langle h \rangle \mapsto \text{SubS}_{S^*}(\tilde{h}')$ is added to $\tilde{\mathcal{T}}$, and thus $\mathcal{A}$'s view is identical in the two hybrids. It follows that

$$\Pr[\text{Hyb}_5^{(b)}(\lambda)] = \Pr[\text{Hyb}_{6,0,\text{int}}^{(b)}(\lambda)]. \quad \square$$

Lemma 3.17. For $L - L_{\text{ct}} = \omega(\log \lambda)$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ and all $\chi \in [q_{\text{ZT}}]$,

$$|\Pr[\text{Hyb}_{6,\chi-1,\text{int}}^{(b)}(\lambda)] - \Pr[\text{Hyb}_{6,\chi}^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. All that differs between the two hybrids is the behavior of the zero-test oracle on the χ^{th} post-challenge query the adversary $\mathcal{A}$ makes. Specifically, for said zero-test query on handle $\langle h \rangle$, the oracle checks if the challenger's formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ is identically zero in hybrid $\text{Hyb}_{6,\chi-1,\text{int}}^{(b)}$, while it instead automatically returns “non-zero” when certain message- and challenge-set-dependent terms are present in the adversary's formal polynomial $\hat{h} = \hat{\mathcal{T}}(\langle h \rangle)$ in hybrid $\text{Hyb}_{6,\chi}^{(b)}$. We will argue that, except with negligible probability, the challenger's formal polynomial $\tilde{h}$ in this case would anyway not be identically zero and the oracle would output “non-zero” in any case. We let bad denote the event that $\tilde{h}$ (corresponding to an $\hat{h}$ that has the challenge message- or challenge-set-dependent terms) is identically zero.

Before continuing, we recall how the challenger's formal polynomials relate to the adversary's in hybrid $\text{Hyb}_{6,\chi-1,\text{int}}^{(b)}$. Let MapP_{int} denote the ring homomorphism defined as MapP , except that $\hat{c}_{4,i}^* \mapsto \tilde{r}_i$ for every $i \in [L]$ (matching the entries $\langle \hat{c}_{4,i}^* \rangle \mapsto \tilde{r}_i$ recorded in the challenge phase of $\text{Hyb}_{6,0,\text{int}}^{(b)}$). By the same inductive argument as in [Lemma 3.14](#), $\tilde{h} = \text{MapP}_{\text{int}}(\hat{h})$ for every handle $\langle h \rangle$ up to the substitution point, which in this hybrid comes right after the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query. From that point onwards, by the inductive argument of [Lemma 3.16](#) (and since SubS_{S^*} is a ring homomorphism applied to every entry), $\tilde{h} = (\text{SubS}_{S^*} \circ \text{MapP}_{\text{int}})(\hat{h})$. Hence, for any χ^{th} post-challenge zero-test query, the challenger's formal polynomial in hybrid $\text{Hyb}_{6,\chi-1,\text{int}}^{(b)}$ can be derived from the adversary's formal polynomial via the following substitutions:

$$\begin{aligned}
\mathbb{G}_1 \quad & \begin{cases} \hat{c}_2^* \mapsto \tilde{s} \\ \hat{w}_i \mapsto \tilde{w}_i & \text{for } i \in [K] \\ \widehat{\tau w}_i \mapsto \tilde{\tau} \tilde{w}_i & \text{for } i \in [K] \\ \hat{y}_i \mapsto \tilde{y}_i & \text{for } i \in [L] \end{cases} & \begin{cases} \hat{c}_{3,i}^* \mapsto \tilde{s}(\tilde{\tau} - \text{id}^*)\tilde{w}_i & \text{for } i \in [K] \\ \hat{c}_{4,i}^* \mapsto \begin{cases} \tilde{s}\tilde{y}_i & i \in S^* \wedge \chi' \geq \chi \\ \tilde{r}_i & \text{otherwise} \end{cases} & \text{for } i \in [L], \end{cases} \\
\mathbb{G}_2 \quad & \begin{cases} \widehat{\tau^j} \mapsto \tilde{\tau}^j & \text{for } j \in [B] \\ \hat{\beta}_{i,\xi} \mapsto \tilde{\beta}_{i,\xi} & \text{for } (i, \xi) \in [L] \times [q_{\text{KG}}] \end{cases} & \hat{\delta}_\xi \mapsto \tilde{\alpha} + \left(\sum_{i \in [K]} y_{i,\xi} \tilde{w}_i \right) \cdot F_\xi(\tilde{\tau}) + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} & \text{for } \xi \in [q_{\text{KG}}], \\
\mathbb{G}_T \quad & \hat{\alpha} \mapsto \tilde{\alpha} & \hat{c}_1^* \mapsto \tilde{s}\tilde{\alpha} + \tilde{m}_b^*.
\end{aligned} \tag{3.3}$$

where $\tilde{m}_b^* = \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ is the challenger's formal polynomial associated with the challenge message handle $\langle m_b^* \rangle$.

We are now ready to go over our main analysis of the χ^{th} zero-test query, which broadly breaks down into three cases. The first two analyze what happens when the adversary “combines” K or fewer, or more than K , keys respectively to break the system, while the third covers remaining edge cases. In more detail (but still at a high level), for an adversary to distinguish the two hybrids, it must query an adversary formal polynomial $\hat{h}$ that maps to an identically zero challenger formal polynomial $\tilde{h}$ where $\hat{h}$ has a $\hat{c}_1^*$ term and/or a $\hat{c}_{4,i}^* \hat{v}$ term for some $\hat{v} \in \{1\} \cup \hat{\mathcal{V}}_2$, per the conditions added in the subsequent hybrid. Below, we provide intuition for which terms in $\hat{h}$ are necessary to keep track of in our analysis (this applies to both the cases of combining greater than K and K or fewer keys).

If $\hat{h}$ has a $\hat{c}_1^*$ term, then by construction this induces $\tilde{s}\tilde{\alpha}$ terms in $\tilde{h}$, which the adversary can attempt to cancel out by including other terms in $\hat{h}$ which induce $\tilde{s}\tilde{\alpha}$ terms in $\tilde{h}$. By inspection of [Eq. \(3.3\)](#) (and as we will justify in more detail later), $\tilde{s}\tilde{\alpha}$ can only otherwise be obtained by pairing $\hat{c}_2^*$ with some $\hat{\delta}_\xi$ (i.e., $\hat{h}$ must include $\hat{c}_2^* \hat{\delta}_\xi$ terms for any cancellation to occur).

Note that the $\hat{c}_2^* \hat{\delta}_\xi$ terms in $\hat{h}$ also induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$, and these can only be cancelled by including *additional* terms in $\hat{h}$ which induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$. As shown in [Eq. \(3.3\)](#) (and as we will justify in more detail a bit later), the only other way to obtain $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ is to pair $\hat{c}_{4,i}^*$ with some $\hat{\beta}_{i,\xi}$ (i.e., $\hat{h}$ must include $\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}$ terms for any cancellation to occur).

For any terms of the form $\hat{c}_{4,i}^* \hat{v}$ for $\hat{v} \in \{1\} \cup \hat{\mathcal{V}}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]}$ (i.e., the other challenge-set-dependent terms), a simple linear independence argument shows that cancellation does not occur if they are present.

To formalize the above intuition, consider the following events based on the adversary's formal polynomial $\hat{h}$:

$$\begin{aligned}
E_1 &:= [\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0], \\
E_2 &:= [\text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0 \text{ for some } (i, j, \xi) \in [L]^2 \times [q_{\text{KG}}]], \\
E_3 &:= [\text{coeff}_{\hat{c}_{4,i}^* \hat{v}}(\hat{h}) \neq 0 \text{ for some } (i, \hat{v}) \in [L] \times (\{1\} \cup \hat{\mathcal{V}}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]})].
\end{aligned}$$

By construction, we have that E_1 is the message-dependent condition and $E_2 \vee E_3$ is the challenge-set-dependent condition, and thus the bad event can only occur in events E_1, E_2 , or E_3 . Next, define $D = \{\xi \in [q_{KG}] : \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \neq 0\}$ and consider the following event that captures when the adversary “combines” K or fewer keys:

$$E_4 := [|D| \leq K].$$

With the above events, we can bound the probability of the bad event as

$$\begin{aligned} \Pr[\text{bad}] &= \Pr[\text{bad} \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge (E_1 \vee E_2 \vee E_3) \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &= \Pr[\text{bad} \wedge (E_1 \vee (\neg E_1 \wedge E_2) \vee E_3) \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge E_1 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4] + \Pr[\text{bad} \wedge E_3 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge E_1 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4] + \Pr[\text{bad} \wedge E_3] + \Pr[\text{bad} \wedge \neg E_4] \\ &= \Pr[\text{bad}_{I-a}] + \Pr[\text{bad}_{I-b}] + \Pr[\text{bad}_{III}] + \Pr[\text{bad}_{II}], \end{aligned}$$

where

$$\begin{aligned} \text{bad}_{I-a} &:= \text{bad} \wedge E_1 \wedge E_4 & \text{bad}_{I-b} &:= \text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4 \\ \text{bad}_{II} &:= \text{bad} \wedge \neg E_4 & \text{bad}_{III} &:= \text{bad} \wedge E_3. \end{aligned}$$

We now analyze the individual bad sub-cases below.

- **Case I: $\hat{h}$ combines K or fewer keys:** We begin by analyzing the two bad events in which $|D| \leq K$. As mentioned before, within this case we will analyze two sub-cases.
 - **Case I-a: $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$:** At a high level, we analyze this case by arguing that a $\hat{c}_1^*$ term in $\hat{h}$ induces $\tilde{s}\tilde{\alpha}$ terms in $\tilde{h}$, which can only be cancelled by $\hat{c}_2^* \hat{\delta}_\xi$ terms in $\hat{h}$ (as these are the only other terms that also induce $\tilde{s}\tilde{\alpha}$ terms in $\tilde{h}$). These $\hat{c}_2^* \hat{\delta}_\xi$ also induce $\tilde{s}\tilde{w}_i$, or $\tilde{s}\tilde{w}_i \tilde{\tau}^j$ terms in $\tilde{h}$. If K or fewer different $\hat{c}_2^* \hat{\delta}_\xi$ terms are present, a cancellation cannot occur between these $\tilde{s}\tilde{w}_i$ or $\tilde{s}\tilde{w}_i \tilde{\tau}^j$ terms.

We now identify which terms (that are not $\hat{c}_1^*$) in $\hat{h}$ can induce $\tilde{s}\tilde{\alpha}$ terms in $\tilde{h}$. From Eq. (3.3), we can see that there are three types of terms in $\hat{h}$ that can induce the variable $\tilde{s}$ in $\tilde{h}$. These are $\hat{c}_2^*$, $\hat{c}_{3,i}^*$ for a subset of $i \in [K]$, and $\hat{c}_{4,j}^*$ for a subset of $j \in [L]$. First observe that $\hat{c}_{4,j}^*$ induces $\tilde{s}\tilde{y}_j$ or $\tilde{r}_j$; in either case, this includes a variable that is not in $\tilde{s}\tilde{\alpha}$, and so $\hat{c}_{4,j}^*$ cannot induce this (even if paired with other variables). Similarly, $\hat{c}_{3,i}^*$ induces $\tilde{s}\tilde{w}_i(\tilde{\tau} - \text{id}^*)$, which again includes a variable that is not in $\tilde{s}\tilde{\alpha}$, and so $\hat{c}_{3,i}^*$ cannot induce this either (even if paired with other variables). Next, note that $\hat{c}_2^* \in \hat{\mathcal{V}}_1$ induces $\tilde{s}$ and thus must be paired with some $\hat{v} \in \hat{\mathcal{V}}_2$ that induces terms of the form $\tilde{\alpha}$. From Eq. (3.3), we can see that $\hat{\delta}_\xi$ for $\xi \in [q_{KG}]$ is the only type of variable in $\hat{\mathcal{V}}_2$ that satisfies this.

Note that $\hat{\delta}_\xi$ also induces $\tilde{w}_i \tilde{\tau}^j$ terms, so each $\hat{c}_2^* \hat{\delta}_\xi$ term in $\hat{h}$ induces $\tilde{s}\tilde{w}_i$ and $\tilde{s}\tilde{w}_i \tilde{\tau}^j$ terms in $\tilde{h}$ in addition to $\tilde{s}\tilde{\alpha}$ terms. As these are the terms we will use to argue that no cancellation can occur when K or fewer different $\hat{c}_2^* \hat{\delta}_\xi$ terms are present, we also identify which other terms in $\hat{h}$ can induce $\tilde{s}\tilde{w}_i$ or $\tilde{s}\tilde{w}_i \tilde{\tau}^j$ terms in $\tilde{h}$. Returning to the three types of terms that induce the variable $\tilde{s}$, we again have that $\hat{c}_{4,j}^*$ induces $\tilde{s}\tilde{y}_j$ or $\tilde{r}_j$, both of which include a variable not in $\tilde{s}\tilde{w}_i$, and so $\hat{c}_{4,j}^*$ cannot induce these terms. The remaining type is $\hat{c}_{3,i}^*$, which induces $\tilde{s}\tilde{w}_i(\tilde{\tau} - \text{id}^*)$ alone and $\tilde{s}\tilde{w}_i \tilde{\tau}^j(\tilde{\tau} - \text{id}^*)$ when paired with $\tilde{\tau}^j \in \hat{\mathcal{V}}_2$ for $j \in [B]$. Pairing $\hat{c}_{3,i}^*$ with any of the remaining variables in $\hat{\mathcal{V}}_2$ (i.e., $\hat{\beta}_{j,\xi}$ or $\hat{\delta}_\xi$) induces an additional variable, and so pairing with these terms cannot induce $\tilde{s}\tilde{w}_i$ or $\tilde{s}\tilde{w}_i \tilde{\tau}^j$ terms either. Altogether, the only terms in $\hat{h}$ that induce $\tilde{s}\tilde{\alpha}$, $\tilde{s}\tilde{w}_i$, or $\tilde{s}\tilde{w}_i \tilde{\tau}^j$ terms in $\tilde{h}$ are $\hat{c}_1^*$, $\hat{c}_2^* \hat{\delta}_\xi$, $\hat{c}_{3,i}^*$, and $\hat{c}_{3,i}^* \tilde{\tau}^j$.

Recalling that $D = \{\xi \in [q_{KG}] : \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \neq 0\}$, we write the challenger’s formal polynomial $\tilde{h}$

corresponding to the adversary's formal polynomial $\hat{h}$ on the χ^{th} query as

$$\begin{aligned} \tilde{h} = & \tilde{\mu} + \text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \tilde{s}\tilde{\alpha} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \left(\tilde{\alpha} + \mathbf{y}_\xi^\top \tilde{\mathbf{w}} \cdot \mathbf{f}_\xi^\top \tilde{\tau} + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) \\ & + \sum_{i \in [K]} \text{coeff}_{\hat{c}_{3,i}^*}(\hat{h}) \cdot \tilde{s}\tilde{\mathbf{w}}_i(\tilde{\tau} - \text{id}^*) + \sum_{(i,j) \in [K] \times [B]} \text{coeff}_{\hat{c}_{3,i}^* \hat{\tau}^j}(\hat{h}) \cdot \tilde{s}\tilde{\mathbf{w}}_i \tilde{\tau}^j (\tilde{\tau} - \text{id}^*), \end{aligned} \quad (3.4)$$

where $\tilde{\mu}$ is a polynomial over $\tilde{\mathcal{V}}$ such that $\text{coeff}_{\tilde{s}\tilde{\alpha}}(\tilde{\mu}) = 0$, $\text{coeff}_{\tilde{s}\tilde{\mathbf{w}}_i}(\tilde{\mu}) = 0$, and $\text{coeff}_{\tilde{s}\tilde{\mathbf{w}}_i \tilde{\tau}^j}(\tilde{\mu}) = 0$ for all $(i, j) \in [K] \times [B + 1]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{s}\tilde{\alpha}$, $\tilde{s}\tilde{\mathbf{w}}_i$, or $\tilde{s}\tilde{\mathbf{w}}_i \tilde{\tau}^j$ terms). This holds because, as argued above, $\hat{c}_1^*$, $\hat{c}_2^* \hat{\delta}_\xi$, $\hat{c}_{3,i}^*$, and $\hat{c}_{3,i}^* \hat{\tau}^j$ are the only terms in $\hat{h}$ that induce $\tilde{s}\tilde{\alpha}$, $\tilde{s}\tilde{\mathbf{w}}_i$, or $\tilde{s}\tilde{\mathbf{w}}_i \tilde{\tau}^j$ terms in $\tilde{h}$. Note that while $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \tilde{m}_b^*$ is absorbed into $\tilde{\mu}$, the above holds because $\tilde{m}_b^*$ contains no $\tilde{s}\tilde{\alpha}$ monomial, due to the fact that $\langle m_b^* \rangle$ must have been created before the challenge phase and $\tilde{s}$ does not appear in any $\tilde{\mathcal{T}}$ entry before then. Next, we define a new polynomial $\tilde{h}_{\text{id}^*}$ that is equal to substituting $\tilde{\tau} = \text{id}^*$ into $\tilde{h}$. We have that

$$\tilde{h}_{\text{id}^*} = \tilde{\mu}' + \text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \tilde{s}\tilde{\alpha} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \left(\tilde{\alpha} + \sum_{i \in [K]} y_{i,\xi} F_\xi(\text{id}^*) \cdot \tilde{\mathbf{w}}_i + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) \quad (3.5)$$

where $\tilde{\mu}'$ is a polynomial over $\tilde{\mathcal{V}}$ such that $\text{coeff}_{\tilde{s}\tilde{\alpha}}(\tilde{\mu}') = 0$ and $\text{coeff}_{\tilde{s}\tilde{\mathbf{w}}_i}(\tilde{\mu}') = 0$ for all $i \in [K]$. Notably, every $\tilde{s}\tilde{\mathbf{w}}_i \tilde{\tau}^j$ term in $\tilde{h}$ maps to a $\tilde{s}\tilde{\mathbf{w}}_i$ term in $\tilde{h}_{\text{id}^*}$, and the former is not present in $\tilde{\mu}$ (as discussed above). Furthermore, while $\tilde{s}\tilde{\alpha}\tilde{\tau}^j$ terms in $\tilde{h}$ map to $\tilde{s}\tilde{\alpha}$ terms in $\tilde{h}_{\text{id}^*}$ (so we must argue that $\tilde{\mu}$ does not contain them), we note that such terms cannot be present in $\tilde{h}$ at all. In particular, recall that $\hat{c}_1^*$ and $\hat{c}_2^* \hat{\delta}_\xi$ are the only terms in $\hat{h}$ that can induce terms in $\tilde{h}$ containing both $\tilde{s}$ and $\tilde{\alpha}$ variables (without extra variables not in $\tilde{s}\tilde{\alpha}\tilde{\tau}^j$). By inspecting Eq. (3.3), we can see that neither induces $\tilde{s}\tilde{\alpha}\tilde{\tau}^j$ terms. Rearranging Eq. (3.5) further, we have that

$$\tilde{h}_{\text{id}^*} = \tilde{\mu}'' + \tilde{s}\tilde{\alpha} \cdot \left(\text{coeff}_{\hat{c}_1^*}(\hat{h}) + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \right) + \sum_{i \in [K]} \tilde{s}\tilde{\mathbf{w}}_i \cdot \left(\sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot y_{i,\xi} F_\xi(\text{id}^*) \right), \quad (3.6)$$

where $\tilde{\mu}'' = \tilde{\mu}' + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \sum_{i \in S_\xi} \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$. Intuitively, we have absorbed the $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms into $\tilde{\mu}''$ so that $\tilde{\mu}''$ contains all the terms that are not core to the analysis of the $|D| \leq K$ case. By construction, we have that $\text{coeff}_{\tilde{s}\tilde{\mathbf{w}}_i}(\tilde{\mu}'') = 0$ for each $i \in [K]$.

Label the elements of $D = \{\xi_1, \dots, \xi_{|D|}\}$. We can further rewrite Eq. (3.6) as $\tilde{h}_{\text{id}^*} = \tilde{\mu}''' + \tilde{s}\tilde{\mathbf{w}}^\top (\mathbf{Y}\mathbf{x})$ for

$$\mathbf{Y} = (\mathbf{y}_{\xi_1}, \dots, \mathbf{y}_{\xi_{|D|}}) \quad \text{and} \quad \mathbf{x} = (\text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi_1}}(\hat{h}) \cdot F_{\xi_1}(\text{id}^*), \dots, \text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi_{|D|}}}(\hat{h}) \cdot F_{\xi_{|D|}}(\text{id}^*)),$$

and where $\tilde{\mu}''' = \tilde{\mu}'' + \tilde{s}\tilde{\alpha} \cdot (\text{coeff}_{\hat{c}_1^*}(\hat{h}) + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}))$. Intuitively, we have absorbed the $\tilde{s}\tilde{\alpha}$ term into $\tilde{\mu}'''$, as this term is not used in the next step (but will come back later).

Observe that if $\mathbf{Y}\mathbf{x} \neq \mathbf{0}$, then $\tilde{s}\tilde{\mathbf{w}}^\top \mathbf{Y}\mathbf{x} \neq 0$. This is because we can express it as a sum $\tilde{s}\tilde{\mathbf{w}}^\top \mathbf{Y}\mathbf{x} = \sum_{i \in [K]} z_i \cdot \tilde{s}\tilde{\mathbf{w}}_i$ where $\mathbf{Y}\mathbf{x} = (z_1, \dots, z_K)$, which is identically zero if and only if $z_i = 0$ for all $i \in [K]$. It follows, in this case, that $\tilde{h}_{\text{id}^*} = \tilde{\mu}''' + \tilde{s}\tilde{\mathbf{w}}^\top \mathbf{Y}\mathbf{x} \neq 0$, because $\text{coeff}_{\tilde{s}\tilde{\mathbf{w}}_i}(\tilde{s}\tilde{\mathbf{w}}^\top \mathbf{Y}\mathbf{x}) \neq 0$ for some $i \in [K]$ while $\text{coeff}_{\tilde{s}\tilde{\mathbf{w}}_i}(\tilde{\mu}''') = 0$ for all $i \in [K]$ as shown above (i.e., $\tilde{s}\tilde{\mathbf{w}}^\top \mathbf{Y}\mathbf{x}$ consists solely of $\tilde{s}\tilde{\mathbf{w}}_1, \dots, \tilde{s}\tilde{\mathbf{w}}_K$ terms and $\tilde{\mu}'''$ does not contain any, so there are no terms in $\tilde{\mu}'''$ that can cancel out terms in $\tilde{s}\tilde{\mathbf{w}}^\top \mathbf{Y}\mathbf{x}$). Furthermore, if $\tilde{h}_{\text{id}^*} \neq 0$, then $\tilde{h} \neq 0$, as $\tilde{h}_{\text{id}^*}$ was obtained by evaluating $\tilde{h}$ on id^* .

Now suppose that $\mathbf{Y}\mathbf{x} = \mathbf{0}$. Recall by the halting condition introduced in hybrid $\text{Hyb}_1^{(b)}$ that any collection of K or fewer different $\mathbf{y}_\xi$ vectors must be linearly independent. Since $|D| \leq K$ in this case, it follows that the columns $\mathbf{y}_{\xi_1}, \dots, \mathbf{y}_{\xi_{|D|}}$ of $\mathbf{Y}$ are linearly independent (i.e., $\text{rank}(\mathbf{Y}) = |D|$), and so the only vector in the kernel of $\mathbf{Y}$ is $\mathbf{0}$. Consequently $\mathbf{x} = \mathbf{0}$ (i.e., every coefficient in $\mathbf{x}$ is 0). Now observe that because the

adversary $\mathcal{A}$ is admissible, then id^* is not a member of any of the sets $I_{\xi_1}, \dots, I_{\xi_{|D|}}$ and thus id^* is not a root of the polynomials $F_{\xi_1}, \dots, F_{\xi_{|D|}}$ and so $F_{\xi}(\text{id}^*) \neq 0$ for $\xi \in D$ (because $\mathcal{I}_{\lambda} = \{0, 1\}^{\lambda}$ and $p > 2^{\lambda}$). This implies that, for every coefficient in $\mathbf{x}$ to be 0, it must hold that $\text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi}}(\hat{h}) = 0$ for all $\xi \in D$; that is, D must be empty. In this case, we can rewrite Eq. (3.6) as

$$\tilde{h}_{\text{id}^*} = \tilde{\mu}'' + \text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \tilde{s} \tilde{\alpha}.$$

As $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$ (per our supposition for this case) and $\text{coeff}_{\tilde{s} \tilde{\alpha}}(\tilde{\mu}'') = 0$ (argued above), then $\tilde{h}_{\text{id}^*} \neq 0$, and thus $\tilde{h} \neq 0$. The zero-test oracle always returns “non-zero” in this case. Thus,

$$\Pr[\text{bad}_{\text{I-a}}] = 0.$$

- **Case I-b: $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = 0$ and $\text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0$ for some $(i, j, \xi) \in [L]^2 \times [q_{\text{KG}}]$:** At a very high level, we analyze this case by arguing that a $\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}$ term in $\hat{h}$ induces $\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$. The former can only be cancelled by including $\hat{c}_2^* \hat{\delta}_{\xi}$ terms in $\hat{h}$ (as these are the only other terms that also induce $\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}$). If K or fewer different $\hat{c}_2^* \hat{\delta}_{\xi}$ terms are present, such a cancellation cannot occur per an argument over $\tilde{s} \tilde{w}_i$ terms similar to Case I-a. For the latter, the lack of cancellation is straightforward but tedious.

We now identify which terms in $\hat{h}$ can induce terms of the form $\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ in $\tilde{h}$. By inspecting Eq. (3.3), we can see that there are two types of terms in $\hat{h}$ that can induce the formal variable $\tilde{\beta}_{j,\xi}$ in $\tilde{h}$. These are $\hat{\beta}_{j,\xi}$ and $\hat{\delta}_{\xi}$ which are both in $\hat{\mathcal{V}}_2$. For the former, it must be paired with some $\hat{v} \in \hat{\mathcal{V}}_1$ that induces terms of the form $\tilde{s} \tilde{y}_i$ or $\tilde{r}_i$. From Eq. (3.3), we can see that $\hat{c}_{4,i}^*$ is the only type of variable in $\hat{\mathcal{V}}_1$ that satisfies this. For the latter, it must be paired with some $\hat{v} \in \hat{\mathcal{V}}_1$ that induces a solo $\tilde{s}$ variable (i.e., without any additional variables multiplied), and from Eq. (3.3) we can see only $\hat{c}_2^*$ satisfies this.

Let $C = \{(i, j, \xi) \in [L]^2 \times [q_{\text{KG}}] : \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0\}$ be the set of indices for $\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}$ and define the following sets:

$$\begin{aligned} C' &= \{(i, \xi) : (i, j, \xi) \in C \wedge i = j\}, & C'' &= \{(i, j, \xi) \in C : i \neq j\}, \\ C'_{S^*} &= \{(i, \xi) \in C' : i \in S^*\}, & C''_{S^*} &= \{(i, j, \xi) \in C'' : i \in S^*\}. \end{aligned}$$

Intuitively, C' and C'' can be thought of as partitioning C into the “diagonal” and “off-diagonal” $\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}$. Furthermore, C'_{S^*} and C''_{S^*} are the subsets that induce $\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$ (because $i \in S^*$) as opposed to $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$.

Like Case I-a, we define a new polynomial $\tilde{h}_{\text{id}^*}$ obtained by substituting in $\tilde{\tau} = \text{id}^*$ into the challenger’s formal polynomial $\tilde{h}$ corresponding to the adversary’s formal polynomial $\hat{h}$ on the χ^{th} query. Recalling that $D = \{\xi \in [q_{\text{KG}}] : \text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi}}(\hat{h}) \neq 0\}$, we can write $\tilde{h}_{\text{id}^*}$ as

$$\begin{aligned} \tilde{h}_{\text{id}^*} &= \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi}}(\hat{h}) \cdot \tilde{s} \left(\tilde{\alpha} + \sum_{i \in [K]} y_{i,\xi} F_{\xi}(\text{id}^*) \cdot \tilde{w}_i + \sum_{i \in S_{\xi}} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) \\ &\quad + \sum_{(i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot (\tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi}) + \sum_{(i,\xi) \in C' \setminus C'_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot (\tilde{r}_i \tilde{\beta}_{i,\xi}) \\ &\quad + \sum_{(i,j,\xi) \in C''_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot (\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}) + \sum_{(i,j,\xi) \in C'' \setminus C''_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot (\tilde{r}_i \tilde{\beta}_{j,\xi}) \end{aligned} \quad (3.7)$$

where $\tilde{\mu}$ is a polynomial over $\hat{\mathcal{V}}$ such that $\text{coeff}_{\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = \text{coeff}_{\tilde{r}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = 0$ for all $(i, j, \xi) \in [L]^2 \times [q_{\text{KG}}]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms). This holds because, as argued above, $\hat{c}_2^* \hat{\delta}_{\xi}$ and $\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}$

are the only terms in $\hat{h}$ that can induce $\tilde{s}\tilde{y}_i\tilde{\beta}_{j,\xi}$ or $\tilde{r}_i\tilde{\beta}_{j,\xi}$ terms, and we do not need to consider the $\tilde{m}_b^*$ term induced by $\hat{c}_1^*$ because we assume that $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = 0$ in this case. Moreover, while $\tilde{s}\tilde{y}_i\tilde{\beta}_{j,\xi}\tilde{\tau}^k$ and $\tilde{r}_i\tilde{\beta}_{j,\xi}\tilde{\tau}^k$ terms in $\tilde{h}$ map to $\tilde{s}\tilde{y}_i\tilde{\beta}_{j,\xi}$ and $\tilde{r}_i\tilde{\beta}_{j,\xi}$ terms in $\tilde{h}_{\text{id}^*}$ (so we must argue that they do not contribute to $\tilde{\mu}$), we note that such terms cannot be present in $\tilde{h}$ at all. In particular, recall that $\hat{\beta}_{j,\xi}$ and $\hat{\delta}_\xi$ are the only terms in $\hat{\mathcal{V}}_2$ that induce $\tilde{\beta}_{j,\xi}$. By inspecting Eq. (3.3), we can see that neither induces $\tilde{\beta}_{j,\xi}\tilde{\tau}^k$ terms, and that every $\hat{v} \in \hat{\mathcal{V}}_1$ inducing a $\tilde{\tau}$ variable (i.e., $\widehat{\tau w_{i'}}$ or $\hat{c}_{3,i'}^*$) also induces an extra $\tilde{w}_{i'}$ variable. Furthermore, $\text{coeff}_{\tilde{s}\tilde{w}_i}(\tilde{\mu}) = 0$ for all $i \in [K]$ (i.e., $\tilde{\mu}$ also does not contain any $\tilde{s}\tilde{w}_i$ terms). This is because, as shown in Case I-a (and also again because $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = 0$), $\hat{c}_{3,i}^*$, and $\hat{c}_2^*\hat{\delta}_\xi$ are the only terms in $\hat{h}$ that induce $\tilde{s}\tilde{w}_i$ terms, and because every $\tilde{s}\tilde{w}_i$ or $\tilde{s}\tilde{w}_i\tilde{\tau}^j$ term induced by $\hat{c}_{3,i}^*$ vanishes upon substituting $\tilde{\tau} = \text{id}^*$, leaving only the $\hat{c}_2^*\hat{\delta}_\xi$ contributions.

By inspection of Eq. (3.7) and using the above fact on $\tilde{\mu}$, we can see that

$$\begin{aligned} \text{coeff}_{\tilde{r}_i\tilde{\beta}_{i,\xi}}(\tilde{h}_{\text{id}^*}) &= \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) && \text{for } (i, \xi) \in C' \setminus C_{S^*}', \\ \text{coeff}_{\tilde{s}\tilde{y}_i\tilde{\beta}_{j,\xi}}(\tilde{h}_{\text{id}^*}) &= \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C_{S^*}'', \\ \text{coeff}_{\tilde{r}_i\tilde{\beta}_{j,\xi}}(\tilde{h}_{\text{id}^*}) &= \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C'' \setminus C_{S^*}', \end{aligned}$$

(i.e., the only cancellation that can occur is on $\hat{c}_{4,i}^*\hat{\beta}_{i,\xi}$ terms in $\hat{h}$ that induce $\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$). Thus, if C_{S^*}' , $C'' \setminus C_{S^*}'$, or $C' \setminus C_{S^*}'$ are non-empty, then $\tilde{h} \neq 0$.

Now suppose that C_{S^*}' , $C'' \setminus C_{S^*}'$, and $C' \setminus C_{S^*}'$ are all empty (note that this implies that C_{S^*}' must be non-empty, as we assume $\text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{j,\xi}}(\hat{h}) \neq 0$ for some $(i, j, \xi) \in [L]^2 \times [q_{\text{KG}}]$ in this case). We can simplify Eq. (3.7) to

$$\tilde{h}_{\text{id}^*} = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \left(\tilde{\alpha} + \sum_{i \in [K]} y_{i,\xi} F_\xi(\text{id}^*) \cdot \tilde{w}_i + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) + \sum_{(i,\xi) \in C_{S^*}'} \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \cdot (\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}). \quad (3.8)$$

Rearranging Eq. (3.8) further, we have that

$$\begin{aligned} \tilde{h}_{\text{id}^*} &= \tilde{\mu} + \sum_{i \in [K]} \tilde{s}\tilde{w}_i \cdot \left(\sum_{\xi \in D} \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \cdot y_{i,\xi} F_\xi(\text{id}^*) \right) + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s}\tilde{\alpha} \\ &\quad + \sum_{(i,\xi) \in U \cap C_{S^*}'} \tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \right) \\ &\quad + \sum_{(i,\xi) \in U \setminus C_{S^*}'} \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi} + \sum_{(i,\xi) \in C_{S^*}' \setminus U} \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \cdot \tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi} \end{aligned} \quad (3.9)$$

where $U = \bigcup_{\xi \in D} \{(i, \xi) : i \in S_\xi\}$ is the set of indices (i, ξ) such that $\xi \in D$ and $i \in S_\xi$ (i.e., the indices that appear in the left-hand sum in Eq. (3.8)).

For the remainder, we will argue that if D is empty, then Eq. (3.9) shows that $\tilde{h}$ is not identically zero; while, on the other hand, if D is non-empty, then we can rewrite Eq. (3.9) in terms of a matrix as was done in Case I-a, which we will then use to show that $\tilde{h}$ is again not identically zero.

Now suppose that D is empty. By inspection of Eq. (3.9) and using the fact C_{S^*}' is non-empty (as argued above), there exists some $(i, \xi) \in C_{S^*}'$ such that

$$\text{coeff}_{\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}}(\tilde{h}_{\text{id}^*}) = \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) = \text{coeff}_{\hat{c}_{4,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \neq 0,$$

and so $\tilde{h} \neq 0$ in this case.

Now suppose D is non-empty. First, we rewrite Eq. (3.9) as

$$\tilde{h}_{\text{id}^*} = \tilde{\mu}' + \sum_{i \in [K]} \tilde{s}\tilde{w}_i \cdot \left(\sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot y_{i,\xi} F_\xi(\text{id}^*) \right), \quad (3.10)$$

where

$$\begin{aligned} \tilde{\mu}' = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s}\tilde{\alpha} + \sum_{(i,\xi) \in U \cap C_{S^*}'} \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \right) \\ + \sum_{(i,\xi) \in U \setminus C_{S^*}'} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi} + \sum_{(i,\xi) \in C_{S^*}' \setminus U} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}. \end{aligned}$$

Notably, because $\tilde{\mu}$ contains no $\tilde{s}\tilde{w}_i$ terms and the absorbed sums consist solely of $\tilde{s}\tilde{\alpha}$ and $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms, we have that $\text{coeff}_{\tilde{s}\tilde{w}_i}(\tilde{\mu}') = 0$ for all $i \in [K]$.

Now, because D is non-empty, there exists some $\xi' \in D$ such that $\text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi'}}(\hat{h}) \neq 0$. Label the elements of $D = \{\xi_1, \dots, \xi_{|D|}\}$ where $\xi_1 = \xi'$. We can further rewrite Eq. (3.10) as $\tilde{h}_{\text{id}^*} = \tilde{\mu}' + \tilde{s}\tilde{\mathbf{w}}^T(\mathbf{Y}\mathbf{x})$ for

$$\mathbf{Y} = (\mathbf{y}_{\xi_1}, \dots, \mathbf{y}_{\xi_{|D|}}) \quad \text{and} \quad \mathbf{x} = (\text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi_1}}(\hat{h}) \cdot F_{\xi_1}(\text{id}^*), \dots, \text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi_{|D|}}}(\hat{h}) \cdot F_{\xi_{|D|}}(\text{id}^*)).$$

The remainder is identical to the matrix argument in Case I-a. In particular, if $\mathbf{Y}\mathbf{x} \neq \mathbf{0}$, then $\tilde{s}\tilde{\mathbf{w}}^T \mathbf{Y}\mathbf{x} \neq 0$ and furthermore $\tilde{h}_{\text{id}^*} = \tilde{\mu}' + \tilde{s}\tilde{\mathbf{w}}^T \mathbf{Y}\mathbf{x} \neq 0$. Finally, if $\tilde{h}_{\text{id}^*} \neq 0$, then $\hat{h} \neq 0$, as $\tilde{h}_{\text{id}^*}$ was obtained by evaluating $\hat{h}$ on id^* . Now, suppose for the sake of contradiction $\mathbf{Y}\mathbf{x} = \mathbf{0}$. By the halting condition introduced in hybrid $\text{Hyb}_1^{(b)}$, any collection of K or fewer different $\mathbf{y}_\xi$ vectors must be linearly independent. Since $|D| \leq K$ in this case $\text{rank}(\mathbf{Y}) = |D|$ and the only vector in the kernel of $\mathbf{Y}$ is $\mathbf{0}$. Consequently $\mathbf{x} = \mathbf{0}$ (i.e., every coefficient in $\mathbf{x}$ is 0). Because the adversary $\mathcal{A}$ is admissible, then id^* is not a member of any of the sets $I_{\xi_1}, \dots, I_{\xi_{|D|}}$ and thus id^* is not a root of the polynomials $F_{\xi_1}, \dots, F_{\xi_{|D|}}$ and so $F_\xi(\text{id}^*) \neq 0$ for $\xi \in D$. Thus, for every coefficient in $\mathbf{x}$ to be 0, it must hold that $\text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) = 0$ for all $\xi \in D$. However, this contradicts the assumption that $\text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi_1}}(\hat{h}) = \text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi'}}(\hat{h}) \neq 0$.

Wrapping up, we have shown that if $\text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0$ for some $(i, j, \xi) \in [L]^2 \times [q_{KG}]$, $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = 0$, and $|D| \leq K$, then there is no condition under which $\tilde{h}$ is identically zero. Thus,

$$\Pr[\text{bad}_{\text{I-b}}] = 0.$$

- **Case II: $\hat{h}$ combines more than K keys:** We now analyze the bad event in which $|D| > K$. As argued for Case I-b, including $\hat{c}_2^* \hat{\delta}_\xi$ terms in $\hat{h}$ induces $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$, which must be cancelled by also including $\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}$ terms in $\hat{h}$ (as these are the only other terms which induce terms of the form $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$). At a high level, we will show that including K different $\hat{c}_2^* \hat{\delta}_\xi$ terms in $\hat{h}$ will induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$ for a sufficient number of $i \in [L]$ such that, with high probability, there is some i' induced such that $i' \notin S^*$. For i' , the $\hat{c}_{4,i'}^* \hat{\beta}_{i',\xi}$ term needed in $\hat{h}$ induces a $\tilde{r}_{i'} \tilde{\beta}_{i',\xi}$ term as opposed to $\tilde{s}\tilde{y}_{i'} \tilde{\beta}_{i',\xi}$ term (because $i' \notin S^*$), and thus cancellation cannot occur and $\tilde{h}$ is not identically zero.

Because the $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms we analyze are identical to those used in Case I-b, we directly proceed with the cancellation argument. Let $C = \{(i, j, \xi) \in [L]^2 \times [q_{KG}] : \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0\}$ be the set of indices for $\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}$ and define the following sets:

$$\begin{aligned} C' &= \{(i, \xi) : (i, j, \xi) \in C \wedge i = j\}, & C'' &= \{(i, j, \xi) \in C : i \neq j\}, \\ C_{S^*}' &= \{(i, \xi) \in C' : i \in S^*\}, & C_{S^*}'' &= \{(i, j, \xi) \in C'' : i \in S^*\}. \end{aligned}$$

As in Case I-b, the challenger's formal polynomial can be written as

$$\begin{aligned}
\tilde{h} = & \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \left(\tilde{\alpha} + \mathbf{y}_\xi^T \tilde{\mathbf{w}} \cdot \mathbf{f}_\xi^T \tilde{\tau} + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) \\
& + \sum_{(i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \left(\tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) + \sum_{(i,\xi) \in C' \setminus C'_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \left(\tilde{r}_i \tilde{\beta}_{i,\xi} \right) \\
& + \sum_{(i,j,\xi) \in C''_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot \left(\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi} \right) + \sum_{(i,j,\xi) \in C'' \setminus C''_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot \left(\tilde{r}_i \tilde{\beta}_{j,\xi} \right)
\end{aligned} \tag{3.11}$$

where $\tilde{\mu}$ is a polynomial over $\mathcal{V}$ such that $\text{coeff}_{\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = \text{coeff}_{\tilde{r}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = 0$ for all $(i, j, \xi) \in [L]^2 \times [q_{KG}]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms).

By inspection of Eq. (3.11) and using the above fact on $\tilde{\mu}$, we have

$$\begin{aligned}
\text{coeff}_{\tilde{r}_i \tilde{\beta}_{i,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) && \text{for } (i, \xi) \in C' \setminus C'_{S^*}, \\
\text{coeff}_{\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C''_{S^*}, \\
\text{coeff}_{\tilde{r}_i \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C'' \setminus C''_{S^*}.
\end{aligned}$$

Thus, if C''_{S^*} , $C'' \setminus C''_{S^*}$, or $C' \setminus C'_{S^*}$ are non-empty, then $\tilde{h} \neq 0$.

Now suppose that C''_{S^*} , $C'' \setminus C''_{S^*}$, and $C' \setminus C'_{S^*}$ are empty. We can simplify Eq. (3.11) to

$$\tilde{h} = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \left(\tilde{\alpha} + \mathbf{y}_\xi^T \tilde{\mathbf{w}} \cdot \mathbf{f}_\xi^T \tilde{\tau} + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) + \sum_{(i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \left(\tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi} \right), \tag{3.12}$$

Rearranging Eq. (3.12) further, we have that

$$\begin{aligned}
\tilde{h} = & \tilde{\mu}' + \sum_{(i,\xi) \in U \cap C'_{S^*}} \tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \right) \\
& + \sum_{(i,\xi) \in U \setminus C'_{S^*}} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi} + \sum_{(i,\xi) \in C'_{S^*} \setminus U} \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi},
\end{aligned} \tag{3.13}$$

where $U = \bigcup_{\xi \in D} \{(i, \xi) : i \in S_\xi\}$ is the set of indices (i, ξ) such that $\xi \in D$ and $i \in S_\xi$ (i.e., the indices that appear in the top sum in Eq. (3.12)), and where

$$\tilde{\mu}' = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \left(\tilde{\alpha} + \mathbf{y}_\xi^T \tilde{\mathbf{w}} \cdot \mathbf{f}_\xi^T \tilde{\tau} \right).$$

Notably, we have absorbed the $\tilde{s} \tilde{\alpha}$ and $\tilde{s} \tilde{\mathbf{w}}_i \tilde{\tau}^j$ terms into $\tilde{\mu}'$ so that $\tilde{\mu}'$ contains all the terms that are not core to the analysis of the $|D| > K$ case. By construction, we have that $\text{coeff}_{\tilde{s} \tilde{y}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}') = 0$ for all $(i, j, \xi) \in [L]^2 \times [q_{KG}]$.

Intuitively, we can see from Eq. (3.13) that the only cancellations can occur in $U \cap C'_{S^*}$. That is, the sets $U \setminus C'_{S^*}$ and $C'_{S^*} \setminus U$ must be empty (which we will formally argue below) and thus $C'_{S^*} = U$ for the bad event to occur. We conclude by arguing that the adversary has a negligible probability of producing an $\hat{h}$ such that $C'_{S^*} = U$.

Proceeding with the argument outlined above, if $U \neq C'_{S^*}$ then there exists some $(i, \xi) \in (U \setminus C'_{S^*}) \cup (C'_{S^*} \setminus U)$. It follows that $\text{coeff}_{\tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi}}(\tilde{h}) = \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \neq 0$ or $\text{coeff}_{\tilde{s} \tilde{y}_i \tilde{\beta}_{i,\xi}}(\tilde{h}) = \text{coeff}_{\hat{c}_{4,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \neq 0$, so $\tilde{h} \neq 0$ in this case.

It remains to bound the probability that $C'_{S^*} = U$, which itself is bounded by the probability that $U \subseteq C'_{S^*}$. First we recall that $C'_{S^*} = \{(i, \xi) \in C' : i \in S^*\}$ and $U = \bigcup_{\xi \in D} \{(i, \xi) : i \in S_\xi\}$. Per the change added in hybrid $\text{Hyb}_{6, \chi-1, \text{int}}^{(b)}$, the set S^* is sampled after the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query, and so no zero-test query is answered between the sampling of S^* and the current query. Because S^* influences $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$, the adversary has not received any information on S^* , so we have that

$$\Pr[\text{bad}_{\Pi}] \leq \Pr[U \subseteq C'_{S^*}] \leq \Pr\left[\bigcup_{\xi \in D} S_\xi \subseteq S^*\right] \leq \left(\frac{L_{\text{ct}}}{L}\right)^{\frac{L}{\kappa}},$$

where the final inequality follows from the halting condition added in hybrid $\text{Hyb}_2^{(b)}$ which asks that there is no collection of sets $S_{\xi_1}, \dots, S_{\xi_K}$ with union size less than or equal to L/κ . Since $|D| > K$ in this case, this condition applies to any K elements of D and yields $|\bigcup_{\xi \in D} S_\xi| > L/\kappa$.

Finally, recall that $L - L_{\text{ct}} = \omega(\log \lambda)$ and $\kappa > 1$. Consider any $c_1 > 0$. Because $L - L_{\text{ct}} = \omega(\log \lambda)$, there exists some $d_1 \in \mathbb{N}$ such that $L - L_{\text{ct}} \geq c_1 \log \lambda$ for all $\lambda \geq d_1$. Let $c = c_1 \log e/\kappa$. We have that,

$$\begin{aligned} \Pr[\text{bad}_{\Pi}] &\leq \left(1 - \frac{L - L_{\text{ct}}}{L}\right)^{\frac{L}{\kappa}} \\ &\leq \exp\left(-\frac{L - L_{\text{ct}}}{\kappa}\right) \\ &\leq \exp\left(-\frac{c_1 \log \lambda}{\kappa}\right) && \text{for } \lambda \geq d_1 \\ &= \lambda^{-c_1 \log e/\kappa} && \text{for } \lambda \geq d_1 \\ &= \lambda^{-c} && \text{for } \lambda \geq d_1. \end{aligned}$$

Note that since $c_1 > 0$ was arbitrary, $c = c_1 \log e/\kappa$ can take any positive value, and so there exists a negligible function $\text{negl}(\cdot)$ such that

$$\Pr[\text{bad}_{\Pi}] = \text{negl}(\lambda).$$

- **Case III: $\text{coeff}_{\hat{c}_{4,i}^* \hat{\nu}}(\hat{h}) \neq 0$ for some $(i, \hat{\nu}) \in [L] \times (\{1\} \cup \hat{\mathcal{V}}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]})$:** This final case considers the remaining challenge-set-dependent terms not covered in the previous cases.

Explicitly, this case occurs if, for $i \in [L]$, $\text{coeff}_{\hat{c}_{4,i}^*}(\hat{h}) \neq 0$, $\text{coeff}_{\hat{c}_{4,i}^* \hat{\tau}^j}(\hat{h}) \neq 0$ for some $j \in [B]$, or $\text{coeff}_{\hat{c}_{4,i}^* \hat{\delta}_\xi}(\hat{h}) \neq 0$ for some $\xi \in [q_{\text{KG}}]$. We consider these as sub-cases and analyze them below:

- **Case III-a: $\text{coeff}_{\hat{c}_{4,i}^*}(\hat{h}) \neq 0$ for some $i \in [L]$:** A $\hat{c}_{4,i}^*$ term in $\hat{h}$ induces an $\tilde{s}\tilde{y}_i$ or $\tilde{r}_i$ term in $\tilde{h}$. We argue that no other terms induce $\tilde{s}\tilde{y}_i$ or $\tilde{r}_i$ terms in $\tilde{h}$, and thus no cancellation occurs in this case. From Eq. (3.3), we have that no other term in $\hat{h}$ induces a term of the form $\tilde{s}\tilde{y}_i$ and thus the only remaining option to induce it is to pair an element $\hat{v}_1 \in \hat{\mathcal{V}}_1$ and $\hat{v}_2 \in \hat{\mathcal{V}}_2$. If $\hat{v}_1$ is $\hat{c}_2^*$, then this induces an $\tilde{s}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains a (solo) $\tilde{y}_i$ term. If $\hat{v}_1$ is $\hat{y}_i$, then this induces a $\tilde{y}_i$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains an $\tilde{s}$ term. If $\hat{v}_1$ is $\hat{w}_{i'}$, $\widehat{\tau w_{i'}}$, or $\hat{c}_{3,i'}^*$ for some $i' \in [K]$, then this induces a term containing $\tilde{w}_{i'}$, which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$) cannot equal $\tilde{s}\tilde{y}_i$. Finally, we have that no other term in $\hat{h}$ induces a (solo) $\tilde{r}_i$. In particular, a $\hat{c}_1^*$ term induces neither an $\tilde{s}\tilde{y}_i$ nor an $\tilde{r}_i$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{s}$ nor any $\tilde{r}_i$ appears in a $\tilde{\mathcal{T}}$ entry before then.
- **Case III-b: $\text{coeff}_{\hat{c}_{4,i}^* \hat{\tau}^j}(\hat{h}) \neq 0$ for some $(i, j) \in [L] \times [B]$:** A $\hat{c}_{4,i}^* \hat{\tau}^j$ term in $\hat{h}$ induces an $\tilde{s}\tilde{y}_i \tilde{\tau}^j$ or $\tilde{r}_i \tilde{\tau}^j$ term in $\tilde{h}$. We argue that no other terms induce these terms in $\tilde{h}$. From Eq. (3.3), we can see that the only remaining option to induce a $\tilde{s}\tilde{y}_i \tilde{\tau}^j$ or $\tilde{r}_i \tilde{\tau}^j$ term, other than by pairing $\hat{c}_{4,i}^*$ with $\widehat{\tau^j}$ itself, is to pair some other element $\hat{v}_1 \in \hat{\mathcal{V}}_1$ with $\hat{v}_2 \in \hat{\mathcal{V}}_2$. If $\hat{v}_1$ is $\hat{c}_2^*$, then this induces an $\tilde{s}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains a (solo) $\tilde{y}_i \tilde{\tau}^j$ term. If $\hat{v}_1$ is $\hat{y}_i$, then this induces a $\tilde{y}_i$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains an $\tilde{s} \tilde{\tau}^j$ term. If $\hat{v}_1$ is $\hat{c}_{4,i'}^*$ for some $i' \neq i$, then this induces an $\tilde{s}\tilde{y}_{i'}$ or $\tilde{r}_{i'}$ term in $\tilde{h}$, which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$)

cannot equal $\tilde{s}\tilde{y}_i\tilde{r}^j$ or $\tilde{r}_i\tilde{r}^j$ since $i' \neq i$. If $\hat{v}_1$ is $\hat{w}_{i'}$, $\widehat{\tau w_{i'}}$, or $\hat{c}_{3,i'}^*$ for some $i' \in [K]$, then this induces a term containing $\hat{w}_{i'}$, which (paired with any $\hat{v}_2 \in \mathcal{V}_2$) cannot equal $\tilde{s}\tilde{y}_i\tilde{r}^j$ or $\tilde{r}_i\tilde{r}^j$. Finally, we have that $\widehat{\tau}$ is the only element of $\mathcal{V}_2$ which induces a (solo) $\tilde{r}^j$ term, so no other pairing can induce the required $\tilde{r}^j$ factor either. In particular, like Case III-a, a $\hat{c}_1^*$ term induces neither an $\tilde{s}\tilde{y}_i\tilde{r}^j$ nor an $\tilde{r}_i\tilde{r}^j$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{s}$ nor any $\tilde{r}_i$ appears in a $\tilde{\mathcal{T}}$ entry before then.

- **Case III-c: $\text{coeff}_{\hat{c}_{4,i}^* \hat{\delta}_\xi}(\hat{h}) \neq 0$ for some $(i, \xi) \in [L] \times [q_{\text{KG}}]$:** For any $j \in S_\xi$, a $\hat{c}_{4,i}^* \hat{\delta}_\xi$ term in $\hat{h}$ induces an $\tilde{s}\tilde{y}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ or $\tilde{r}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ term in $\tilde{h}$. We argue that no other terms induce these terms in $\tilde{h}$. From Eq. (3.3), we can see that the only remaining option to induce a $\tilde{s}\tilde{y}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ or $\tilde{r}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ term, other than by pairing $\hat{c}_{4,i}^*$ with $\hat{\delta}_\xi$ itself, is to pair some other element $\hat{v}_1 \in \mathcal{V}_1$ with $\hat{v}_2 \in \mathcal{V}_2$. If $\hat{v}_1$ is $\hat{c}_2^*$, then this induces an $\tilde{s}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \mathcal{V}_2$ contains a (solo) $\tilde{y}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ term. If $\hat{v}_1$ is $\hat{y}_i$ or $\hat{y}_j$, then this induces a $\tilde{y}_i$ or $\tilde{y}_j$ term in $\tilde{h}$, however no $\hat{v}_2 \in \mathcal{V}_2$ contains an $\tilde{s}$ or $\tilde{r}_i$ term. If $\hat{v}_1$ is $\hat{c}_{4,i'}^*$ for some $i' \neq i$, then this induces an $\tilde{s}\tilde{y}_{i'}$ or $\tilde{r}_{i'}$ term in $\tilde{h}$, and the only $\hat{v}_2 \in \mathcal{V}_2$ that supplies a $\tilde{\beta}_{j,\xi}$ factor together with a $\tilde{y}$ factor is $\hat{\delta}_\xi$. Pairing the two yields only terms of the form $\tilde{s}\tilde{y}_{i'}\tilde{y}_k\tilde{\beta}_{k,\xi}$ or $\tilde{r}_{i'}\tilde{y}_k\tilde{\beta}_{k,\xi}$ for $k \in S_\xi$. To match the $\tilde{\beta}_{j,\xi}$ factor we need $k = j$, and then the remaining $\tilde{y}_{i'}$ (resp. $\tilde{r}_{i'}$) factor would have to equal $\tilde{y}_i$ (resp. $\tilde{r}_i$), which forces $i' = i$. In particular, this rules out $i' = j$ as pairing $\hat{c}_{4,j}^*$ with $\hat{\delta}_\xi$ gives $\tilde{s}\tilde{y}_j\tilde{y}_i\tilde{\beta}_{i,\xi}$ (when $i \in S_\xi$), which carries $\tilde{\beta}_{i,\xi}$ rather than $\tilde{\beta}_{j,\xi}$ and is therefore a different monomial. If $\hat{v}_1$ is $\hat{w}_{i'}$, $\widehat{\tau w_{i'}}$, or $\hat{c}_{3,i'}^*$ for some $i' \in [K]$, then this induces a term containing $\hat{w}_{i'}$, which (paired with any $\hat{v}_2 \in \mathcal{V}_2$) cannot equal $\tilde{s}\tilde{y}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ or $\tilde{r}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$. Finally, as argued in Case I-b, $\hat{\beta}_{j,\xi}$ and $\hat{\delta}_\xi$ are the only elements of $\mathcal{V}_2$ which induce a $\tilde{\beta}_{j,\xi}$ term, and the former would have to be paired with an element of $\mathcal{V}_1$ inducing $\tilde{s}\tilde{y}_i\tilde{y}_j$ or $\tilde{r}_i\tilde{y}_j$, of which there are none, so no other pairing can induce the required $\tilde{y}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ factor either. We note in particular that $\tilde{\beta}_{j,\xi}$ is unique to the key ξ , so these terms cannot be cancelled by $\hat{c}_{4,i}^* \hat{\delta}_{\xi'}$ terms for any $\xi' \neq \xi$. Note that, like Case III-a, a $\hat{c}_1^*$ term induces neither an $\tilde{s}\tilde{y}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ nor an $\tilde{r}_i\tilde{y}_j\tilde{\beta}_{j,\xi}$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{s}$ nor any $\tilde{r}_i$ appears in a $\tilde{\mathcal{T}}$ entry before then.

Summarizing the analysis above, we have

$$\begin{aligned}
\text{coeff}_{\tilde{s}\tilde{y}_i}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^*}(\hat{h}) && \text{for } i \in S^* \\
\text{coeff}_{\tilde{r}_i}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^*}(\hat{h}) && \text{for } i \in [L] \setminus S^* \\
\text{coeff}_{\tilde{s}\tilde{y}_i\tilde{r}^j}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^* \hat{r}^j}(\hat{h}) && \text{for } (i, j) \in S^* \times [B], \\
\text{coeff}_{\tilde{r}_i\tilde{r}^j}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^* \hat{r}^j}(\hat{h}) && \text{for } (i, j) \in ([L] \setminus S^*) \times [B], \\
\text{coeff}_{\tilde{s}\tilde{y}_i\tilde{y}_j\tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^* \hat{\delta}_\xi}(\hat{h}) && \text{for } (i, \xi) \in S^* \times [q_{\text{KG}}], j \in S_\xi, \\
\text{coeff}_{\tilde{r}_i\tilde{y}_j\tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{4,i}^* \hat{\delta}_\xi}(\hat{h}) && \text{for } (i, \xi) \in ([L] \setminus S^*) \times [q_{\text{KG}}], j \in S_\xi.
\end{aligned}$$

It follows that if any of the Case III terms are included in $\hat{h}$, then $\tilde{h}$ has a corresponding term and thus $\tilde{h}$ is never identically zero. Thus,

$$\Pr[\text{bad}_{\text{III}}] = 0.$$

Combining the case analysis above, we overall have that

$$\begin{aligned}
\Pr[\text{bad}] &\leq \Pr[\text{bad}_{\text{I-a}}] + \Pr[\text{bad}_{\text{I-b}}] + \Pr[\text{bad}_{\text{II}}] + \Pr[\text{bad}_{\text{III}}] \\
&= \text{negl}(\lambda).
\end{aligned}$$

□

Lemma 3.18. For all $\lambda \in \mathbb{N}$ and all $\chi \in [q_{\text{ZT}}]$, $\Pr[\text{Hyb}_{6,\chi}^{(b)}(\lambda)] = \Pr[\text{Hyb}_{6,\chi,\text{int}}^{(b)}(\lambda)]$.

Proof. All that differs between the two hybrids is when the challenger samples the set S^* and performs the associated substitution on the entries of $\tilde{\mathcal{T}}$. Specifically, this step is performed immediately after the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query in hybrid $\text{Hyb}_{6,\chi}^{(b)}$, while it is instead performed immediately after the χ^{th} post-challenge zero-test query in hybrid $\text{Hyb}_{6,\chi,\text{int}}^{(b)}$. We will argue that $\mathcal{A}$'s view is identical in both hybrids.

Throughout, we let Sub_{S^*} denote the ring homomorphism as defined in hybrid $\text{Hyb}_{6,0,\text{int}}^{(b)}$, so that the substitution step performed in both hybrids replaces each entry $\tilde{h}$ of $\tilde{\mathcal{T}}$ with $\text{Sub}_{S^*}(\tilde{h})$. We also recall that, since hybrid $\text{Hyb}_5^{(b)}$,

the zero-test oracle answers a query on a handle $\langle h \rangle$ by testing whether $\tilde{h} \equiv 0$, and so the set S^* sampled in this step influences $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$.

We first observe that the first $\chi - 1$ post-challenge zero-test queries are answered identically in both hybrids, as the substitution step is performed after the $(\chi - 1)^{\text{th}}$ query in either case.

Next, consider the χ^{th} post-challenge zero-test query on a handle $\langle h \rangle$ with associated adversary formal polynomial $\hat{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ and challenger formal polynomial $\tilde{h}' = \tilde{\mathcal{T}}(\langle h \rangle)$ (where the latter is prior to any substitution). As the behavior added in hybrid $\text{Hyb}_{6,\chi}^{(b)}$ applies to this query, if $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$ or $\text{coeff}_{\hat{c}_{4,i}^*}(\hat{h}) \neq 0$ for some $(i, \hat{c}) \in [L] \times (\{1\} \cup \hat{\mathcal{V}}_2)$, then both hybrids return “non-zero” without accessing the table $\tilde{\mathcal{T}}$ at all, and the query is answered identically. Otherwise, $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = \text{coeff}_{\hat{c}_{4,i}^*}(\hat{h}) = 0$ for all $(i, \hat{c}) \in [L] \times (\{1\} \cup \hat{\mathcal{V}}_2)$. By inspection of Eq. (3.3), $\hat{c}_{4,i}^*$ are the only variables in $\hat{\mathcal{V}}$ that can induce a $\tilde{r}_i$ variable in $\tilde{h}'$, and thus $\tilde{h}'$ contains no monomial involving any of the variables $\{\tilde{r}_i\}_{i \in [L]}$. It follows that $\text{SubS}_{S^*}(\tilde{h}') = \tilde{h}'$, and so the χ^{th} query is again answered identically whether or not the substitution has already been applied.

It remains to argue that the entries of $\tilde{\mathcal{T}}$ coincide in both hybrids from the point immediately following the χ^{th} query onwards, from which it follows that the remaining post-challenge zero-test queries are also answered identically. Consider any handle $\langle h \rangle$ present in $\tilde{\mathcal{T}}$ at this point and let $\tilde{h}'$ again denote the entry it would have in the absence of any substitution. In hybrid $\text{Hyb}_{6,\chi,\text{int}}^{(b)}$, the entry is $\text{SubS}_{S^*}(\tilde{h}')$, so it suffices to show that the same holds in hybrid $\text{Hyb}_{6,\chi}^{(b)}$, which we do by induction on the order in which handles are created.

For the base case, a handle created before the substitution step has entry $\text{SubS}_{S^*}(\tilde{h}')$, as the step is applied to every handle in $\tilde{\mathcal{T}}$ at that time. For the inductive step, consider a handle $\langle h \rangle$ created after the substitution point and suppose that every handle $\langle h_0 \rangle$ created before it has entry $\text{SubS}_{S^*}(\tilde{h}_0')$. We go over each way in which $\langle h \rangle$ may have been produced below:

- **Post-challenge key-generation queries:** On the ξ^{th} key-generation query, the challenger adds the entries $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ for each $i \in S_\xi$ and $\langle \delta_\xi \rangle \mapsto \tilde{\alpha} + (\sum_{i \in [K]} y_{i,\xi} \tilde{w}_i) \cdot F_\xi(\tilde{r}) + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi}$ to $\tilde{\mathcal{T}}$. Neither entry contains any of the variables $\{\tilde{r}_i\}_{i \in [L]}$ and so both are unaffected by the SubS_{S^*} ring homomorphism.
- **Post-challenge generic group oracle queries:** The analysis of the generic group oracle queries is as follows:
 - $\text{BGGM.Encode}(x, v)$: The challenger adds the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, which is again unaffected by the SubS_{S^*} ring homomorphism.
 - $\text{BGGM.Add}(\langle h_1 \rangle, \langle h_2 \rangle)$: Recall that the challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that SubS_{S^*} is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{S^*}(\tilde{h}_1') + \text{SubS}_{S^*}(\tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}_1' + \tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}').$$

- $\text{BGGM.Pair}(\langle h_1 \rangle, \langle h_2 \rangle)$: The challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that SubS_{S^*} is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{S^*}(\tilde{h}_1') \cdot \text{SubS}_{S^*}(\tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}_1' \cdot \tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}').$$

In each case, the entry of $\langle h \rangle$ coincides with $\text{SubS}_{S^*}(\tilde{h}')$, and thus $\mathcal{A}$'s view is identical in the two hybrids. It follows that

$$\Pr[\text{Hyb}_{6,\chi}^{(b)}(\lambda)] = \Pr[\text{Hyb}_{6,\chi,\text{int}}^{(b)}(\lambda)]. \quad \square$$

Lemma 3.19. For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_{6,q_{\text{ZT}},\text{int}}^{(0)}(\lambda)] = \Pr[\text{Hyb}_{6,q_{\text{ZT}},\text{int}}^{(1)}(\lambda)]$.

Proof. The lemma follows from the fact that both hybrids operate independently of the challenge bit b . In particular, any zero-test query dependent on the challenge-message-dependent component (which is the only component dependent on the bit b) must have that $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$, and the oracle only ever returns “non-zero” in this case (as the forced-“non-zero” rule applies to every post-challenge zero-test query, and no pre-challenge query can involve $\hat{c}_1^*$). $\square$

The theorem follows by applying a standard hybrid argument over [Lemmas 3.11](#) to [3.19](#). In particular,

$$\begin{aligned} |\Pr[b^* = 1 : b = 0] - \Pr[b^* = 1 : b = 1]| &= |\Pr[\text{Hyb}_0^{(0)}(\lambda)] - \Pr[\text{Hyb}_0^{(1)}(\lambda)]| \\ &\leq 2(4 + q_{\text{ZT}}) \cdot \text{negl}(\lambda) \\ &= \text{negl}(\lambda), \end{aligned}$$

where we have used the fact that $q_{\text{ZT}} = \text{poly}(\lambda)$. $\square$

Parameter instantiation. We now give one example of how to set the parameters of [Construction 3.7](#) so that [Theorem 3.8](#) yields inverse-polynomial error and the conditions of [Theorem 3.10](#) are satisfied. We stress that there is nothing special about our parameter selection; any choice of L , L_{sk} , L_{ct} , K , and P satisfying the conditions suffices. Take

$$L = K = \lambda, \quad L_{\text{sk}} = \lambda^{\frac{1}{3}}, \quad L_{\text{ct}} = \lambda - \lambda^{\frac{1}{3}}, \quad P = \lambda \log^2 \lambda.$$

Per [Theorem 3.8](#), we have that [Construction 3.7](#) is statistically correct with error

$$\varepsilon(\lambda) = L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L}\right) = \lambda^{\frac{1}{3}} \cdot \frac{\lambda^{\frac{1}{3}}}{\lambda} = \lambda^{-\frac{1}{3}}.$$

Furthermore, L and K are polynomial in λ , $L_{\text{sk}} = \lambda^{\frac{1}{3}} = \omega(\log \lambda)$, and $L - L_{\text{ct}} = \lambda^{\frac{1}{3}} = \omega(\log \lambda)$. Moreover, we have that $K \cdot L_{\text{sk}} = \lambda^{\frac{4}{3}} \geq 2\lambda = \kappa \cdot L$ with $\kappa = 2$ (for all $\lambda \geq 8$), and $P = \lambda \log^2 \lambda = \omega(\lambda \log \lambda) = \omega(K \cdot \log \lambda)$ with $P \geq \lambda$ (for all $\lambda \geq 2$). Thus, by [Theorem 3.10](#), [Construction 3.7](#) is semantically secure in the generic bilinear group model.

Lifting to full correctness. As discussed in [Remark 3.4](#), we can reduce [Construction 3.7](#)'s inverse-polynomial error to negligible correctness error. Applying this with the above parameter instantiation, we obtain the following.

Corollary 3.20 (Tagless Batched IBE from Pairings in the BGGM). *Let λ be a security parameter and let GroupGen be a prime-order asymmetric pairing-group generator. There exists a tagless batched identity-based encryption scheme with identity space $\mathcal{I} = \{\mathcal{I}_\lambda\}_{\lambda \in \mathbb{N}}$ where $\mathcal{I}_\lambda = \{0, 1\}^\lambda$ that is correct with negligible error and semantically secure in the generic bilinear group model.*

4 Multi-Authority Attribute-Based Encryption without Random Oracles

Multi-authority attribute-based encryption (MA-ABE) extends ABE to independent authorities that each issue keys over the attributes they manage [[LW11](#)]. Existing pairing-based schemes rely on a hash function that is modeled as a random oracle in security analysis [[LW11](#), [RW15](#), [OT20](#), [DKW23a](#), [DKW23b](#), [DP23](#)]. In this section, we apply our technique to obtain an MA-ABE scheme supporting conjunction policies that does not rely on random oracles.³

4.1 Multi-Authority ABE Definitions

We start by recalling the syntax and security notions for (fully decentralized) multi-authority attribute-based encryption [[LW11](#)]. We consider fully-adaptive security [[DKW23b](#)].

Definition 4.1 (Multi-Authority ABE [[LW11](#), adapted]). A *multi-authority attribute-based encryption* (MA-ABE) scheme with global identifier space $\mathcal{G} = \{\mathcal{G}_\lambda\}_{\lambda \in \mathbb{N}}$, attribute universe $\mathcal{U} = \{\mathcal{U}_\lambda\}_{\lambda \in \mathbb{N}}$, and policy space $\mathcal{P} = \{\mathcal{P}_{M,\lambda}\}_{M,\lambda \in \mathbb{N}}$ is a tuple of efficient algorithms $\Pi_{\text{MA-ABE}} = (\text{GlobalSetup}, \text{AuthSetup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ with the following syntax:

³Strictly speaking, our scheme is proven secure in (a model that implies) the Shoup generic group model [[Sho97](#), [BB04b](#)], which implies the random oracle model [[ZZ23](#)]. However, our scheme makes only black-box use of the group and does not rely on an explainable oblivious sampler, unlike schemes that substantively use the random oracle. This gives a meaningful sense in which our construction avoids random oracles.

- $\text{GlobalSetup}(1^\lambda, 1^M) \rightarrow \text{pp}$: On input the security parameter λ and a policy-size bound M , the global setup algorithm outputs parameters pp . We assume that pp implicitly contains the security parameter 1^λ , policy-size bound 1^M and a description of the message space $\mathcal{M}_\lambda$ (where $|\mathcal{M}_\lambda| \geq 2$).
- $\text{AuthSetup}(\text{pp}, a) \rightarrow (\text{mpk}_a, \text{msk}_a)$: On input the parameters pp and an attribute $a \in \mathcal{U}_\lambda$, the authority-setup algorithm outputs an authority (master) public key mpk_a and authority (master) secret key msk_a . We assume that mpk_a and msk_a implicitly contain the parameters pp and the attribute a , and that one can efficiently check if mpk_a is well-formed (e.g., consisting of elements in the correct groups).
- $\text{KeyGen}(\text{msk}_a, \text{gid}) \rightarrow \text{sk}_{a,\text{gid}}$: On input an authority secret key msk_a and a global identifier $\text{gid} \in \mathcal{G}_\lambda$, the key-generation algorithm outputs a secret key $\text{sk}_{a,\text{gid}}$.
- $\text{Encrypt}(\{\text{mpk}_a\}_{a \in A}, P, m) \rightarrow \text{ct}$: On input a set of authority public keys $\{\text{mpk}_a\}_{a \in A}$ for a set of attributes A , a policy $P \in \mathcal{P}_{M,\lambda}$ over A , and a message $m \in \mathcal{M}_\lambda$, the (randomized) encryption algorithm outputs a ciphertext ct .
- $\text{Decrypt}(\{\text{sk}_{a,\text{gid}}\}_{a \in A}, \text{ct}) \rightarrow m$: On input a collection of secret keys $\{\text{sk}_{a,\text{gid}}\}_{a \in A}$ for a set of attributes A and a fixed identifier $\text{gid} \in \mathcal{G}_\lambda$, and a ciphertext ct , the decryption algorithm outputs a message $m \in \mathcal{M}_\lambda$ or $\perp$.

Remark 4.2 (Multi-Attribute Authorities). Following [RW15], we simplify our syntax and definitions by associating each authority with a *single* attribute. To handle multiple attributes, an authority can simply internally maintain several single-attribute authorities.

Definition 4.3 (Correctness). An MA-ABE scheme $\Pi_{\text{MA-ABE}}$ is *statistically correct* with error $\varepsilon(\lambda)$ if, for all $\lambda, M \in \mathbb{N}$, all parameters pp in the support of $\text{GlobalSetup}(1^\lambda, 1^M)$, all sets of attributes $A \subseteq \mathcal{U}_\lambda$, all authority key-pairs $(\text{mpk}_a, \text{msk}_a)$ in the support of $\text{AuthSetup}(\text{pp}, a)$ for $a \in A$, all policies $P \in \mathcal{P}_{M,\lambda}$ over A , all attribute sets $A' \subseteq A$ satisfying P , all global identifiers $\text{gid} \in \mathcal{G}_\lambda$, all secret keys $\text{sk}_{a,\text{gid}}$ in the support of $\text{KeyGen}(\text{msk}_a, \text{gid})$ for $a \in A'$, and all messages $m \in \mathcal{M}_\lambda$ (where $\mathcal{M}_\lambda$ is the message space associated with pp),

$$\Pr [\text{Decrypt}(\{\text{sk}_{a,\text{gid}}\}_{a \in A'}, \text{ct}) = m : \text{ct} \leftarrow \text{Encrypt}(\{\text{mpk}_a\}_{a \in A}, P, m)] \geq 1 - \varepsilon(\lambda).$$

Remark 4.4 (Negligible Correctness Error). Like the batched IBE setting discussed in Remark 3.4, our definition allows the correctness error $\varepsilon = \varepsilon(\lambda)$ to be non-negligible. When $\varepsilon(\lambda) \leq 1/2 - 1/\text{poly}(\lambda)$, we can achieve *negligible correctness error* by performing a $\text{poly}'(\lambda)$ -wise repetition of the scheme. Concretely, a sender produces $\text{poly}'(\lambda) = \omega(\log \lambda) \cdot \text{poly}(\lambda)^2$ ciphertexts encrypting the same message, and the receiver decrypts all of them and outputs the majority result.

Definition 4.5 (Semantic Security). For an MA-ABE scheme $\Pi_{\text{MA-ABE}}$, an adversary $\mathcal{A}$, a bit $b \in \{0, 1\}$, and a security parameter $\lambda \in \mathbb{N}$, we define the semantic security experiment as follows:

1. **Setup phase:** The adversary $\mathcal{A}$ sends a policy-size bound 1^M to the challenger. In response, the challenger samples $\text{pp} \leftarrow \text{GlobalSetup}(1^\lambda, 1^M)$ and gives pp to $\mathcal{A}$.
2. **Pre-challenge query phase:** The adversary $\mathcal{A}$ can now make the following queries:
 - **Authority-setup queries:** In an authority-setup query, $\mathcal{A}$ specifies an attribute $a \in \mathcal{U}_\lambda$. In response, the challenger samples $(\text{mpk}_a, \text{msk}_a) \xleftarrow{\mathbb{R}} \text{AuthSetup}(\text{pp}, a)$ and gives mpk_a to $\mathcal{A}$.
 - **Authority-corruption queries:** In an authority-corruption query, $\mathcal{A}$ specifies an attribute $a \in \mathcal{U}_\lambda$. In response, the challenger gives msk_a to $\mathcal{A}$.
 - **Key-generation queries:** In a key-generation query, $\mathcal{A}$ specifies an attribute $a \in \mathcal{U}_\lambda$ and a global identifier $\text{gid} \in \mathcal{G}_\lambda$. In response, the challenger samples $\text{sk}_{a,\text{gid}} \xleftarrow{\mathbb{R}} \text{KeyGen}(\text{msk}_a, \text{gid})$ and gives $\text{sk}_{a,\text{gid}}$ to $\mathcal{A}$.
3. **Challenge phase:** At some point, $\mathcal{A}$ outputs a challenge policy $P^* \in \mathcal{P}_{M,\lambda}$ over attributes $A^* \subseteq \mathcal{U}_\lambda$ together with two challenge messages $m_0^*, m_1^* \in \mathcal{M}_\lambda$. It also outputs a well-formed authority public key mpk_{a^*} for each corrupt attribute $a^* \in A^* \setminus A$ where A denotes the set of attributes queried up to this point to the authority-setup oracle. The challenger replies with $\text{ct}^* \leftarrow \text{Encrypt}(\{\text{mpk}_a\}_{a \in A^* \cap A} \cup \{\text{mpk}_{a^*}\}_{a^* \in A^* \setminus A}, P^*, m_b^*)$.

4. **Post-challenge query phase:** Now, $\mathcal{A}$ can continue to make authority setup, authority corruption, and key-generation queries as in the pre-challenge query phase.
5. **Output phase:** At the end of the experiment, $\mathcal{A}$ outputs a bit $b^* \in \{0, 1\}$, which is the output of the experiment.

We let A_{gid} denote the set of attributes queried to the key-generation oracle for a given global identifier $\text{gid} \in \mathcal{G}_\lambda$, and $C \subseteq \mathcal{U}_\lambda$ denote the set of *corrupt* attributes (i.e., the attributes a on which $\mathcal{A}$ made an authority-corruption query, together with the attributes a^* for which $\mathcal{A}$ supplied its own authority public key mpk_{a^*} in the challenge phase). We say that an adversary $\mathcal{A}$ is *admissible* if the following holds:

- For every global identifier $\text{gid} \in \mathcal{G}_\lambda$, the set $(A_{\text{gid}} \cup C) \cap A^*$ does not satisfy the challenge policy P^* .
- No authority-setup query is made on an attribute $a \in \mathcal{U}_\lambda$ that was previously queried to the authority-setup oracle or supplied by the adversary during the challenge phase.
- No corruption or key-generation queries are made on attributes $a \in \mathcal{U}_\lambda$ that were not previously queried to the authority-setup oracle.

We furthermore say that $\Pi_{\text{MA-ABE}}$ is *semantically secure* if, for every efficient admissible adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$|\Pr[b^* = 1 : b = 0] - \Pr[b^* = 1 : b = 1]| \leq \text{negl}(\lambda).$$

Remark 4.6 (Unique-Attribute-Identity Security). [Definition 4.5](#) allows an adversary to make several key-generation queries to the same authority on a single identity, potentially receiving a different identity secret key on each query. We also consider security allowing only one key-generation query per identity to each authority, which we refer to as *unique-attribute-identity semantic security*. It is straightforward to upgrade unique-attribute-identity security to full security by having each authority sample a key for a pseudorandom function (PRF), which it stores in its master secret key and uses to derive the key-generation randomness.

4.2 Multi-Authority ABE Construction

In this section, we describe our construction of multi-authority attribute-based encryption without random oracles. As described in [Section 1.1](#), we obtain our scheme by augmenting a variant of the Lewko-Waters [\[LW11\]](#) multi-authority ABE scheme (where the random oracle is replaced by a K -wise independent hash function) with our slot mechanism (see also [Fig. 1](#)). We provide relevant background on linear secret sharing with linear independence for unauthorized rows in [Section 2](#), and we give our full construction below:

Construction 4.7 (MA-ABE). Let λ be a security parameter, and M be a policy-size bound, and define the slot count $L = L(\lambda, M)$, the number of slots $L_{\text{sk}} = L_{\text{sk}}(\lambda, M)$ associated with each secret key, number of slots $L_{\text{ct}} = L_{\text{ct}}(\lambda, M)$ associated with each ciphertext, and the hash function K -wise independence size $K = K(\lambda, M)$. Let $\mathcal{U} = \{\mathcal{U}_\lambda\}_{\lambda \in \mathbb{N}}$ be an attribute universe and $\mathcal{P} = \{\mathcal{P}_{M,\lambda}\}_{M,\lambda \in \mathbb{N}}$ be a policy space where $\mathcal{P}_{M,\lambda}$ consists of policies that can be described by a linear secret sharing scheme of a size at most M for $\mathcal{U}_\lambda$. Let GroupGen be a prime-order asymmetric pairing-group generator. We construct an MA-ABE scheme $\Pi_{\text{MA-ABE}} = (\text{GlobalSetup}, \text{AuthSetup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ with attribute universe $\mathcal{U}$, policy space $\mathcal{P}$, and global identifier space $\mathcal{G} = \{\mathcal{G}_\lambda\}_{\lambda \in \mathbb{N}}$ for $\mathcal{G}_\lambda = \{0, 1\}^\lambda$ as follows:

- **GlobalSetup**($1^\lambda, 1^M$): On input the security parameter λ and policy-size bound M , the global setup algorithm proceeds as follows:
 - Sample a prime-order asymmetric pairing group $\bar{\mathcal{G}} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \bar{g}_1, \bar{g}_2, e) \leftarrow \text{GroupGen}(1^\lambda)$ and generators $g_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1 \setminus \{\bar{g}_1^0\}$ and $g_2 \xleftarrow{\mathbb{R}} \mathbb{G}_2 \setminus \{\bar{g}_2^0\}$. Define $[1]_1 := g_1$, $[1]_2 := g_2$, and $\mathcal{G} = (\bar{\mathcal{G}}, [1]_1, [1]_2)$.
 - Sample group elements $[u_1]_2, \dots, [u_K]_2 \xleftarrow{\mathbb{R}} \mathbb{G}_2$ and define the function $H(X) := \sum_{i \in [K]} X^{i-1} [u_i]_2$. Note that $H: \mathbb{Z}_p \rightarrow \mathbb{G}_2$ is a K -wise independent hash function keyed by $([u_1]_2, \dots, [u_K]_2)$.

Output the global parameters $\text{pp} = (\mathcal{G}, H)$ and define the message space to be $\mathcal{M}_\lambda := \mathbb{G}_T$.

- **AuthSetup**(pp, a): On input the parameters pp = ($\mathcal{G}, H$) and an attribute $a \in \mathcal{U}_\lambda$, the authority-setup algorithm proceeds by sampling exponents $\alpha_a, y_a, \gamma_{1,a}, \dots, \gamma_{L,a} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and then outputs the authority (master) public key $\text{mpk}_a = ([\alpha_a]_\tau, [y_a]_1, \{[\gamma_{i,a}]_1\}_{i \in [L]})$ and authority (master) secret key $\text{msk}_a = (\alpha_a, y_a, \{\gamma_{i,a}\}_{i \in [L]})$.
- **KeyGen**(msk_a, gid): On input an authority secret key $\text{msk}_a = (\alpha_a, y_a, \{\gamma_{i,a}\}_{i \in [L]})$ and a global identifier $\text{gid} \in \mathcal{G}_\lambda$, the key-generation algorithm proceeds as follows:

- Sample a set $S_{a,\text{gid}} \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$ and sample exponents $\beta_{i,a,\text{gid}} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ for all $i \in S_{a,\text{gid}}$.
- Compute $[\delta_{a,\text{gid}}]_2 = [\alpha_a]_2 + y_a[h_{\text{gid}}]_2 + \sum_{i \in S_{a,\text{gid}}} [\gamma_{i,a}\beta_{i,a,\text{gid}}]_2$ where $[h_{\text{gid}}]_2 \leftarrow H(\text{gid})$.

Output the secret key $\text{sk}_{a,\text{gid}} = ([\delta_{a,\text{gid}}]_2, \{[\beta_{i,a,\text{gid}}]_2\}_{i \in S_{a,\text{gid}}})$.

- **Encrypt**($\{\text{mpk}_a\}_{a \in A}, (\mathbf{M}, \rho), [m]_\tau$): On input a collection of authority public keys $\{\text{mpk}_a\}_{a \in A}$ for a set of attributes A where each $\text{mpk}_a = ([\alpha_a]_\tau, [y_a]_1, \{[\gamma_{i,a}]_1\}_{i \in [L]})$, a policy $(\mathbf{M}, \rho)$ consisting of a share-generating matrix $\mathbf{M} \in \mathbb{Z}_p^{\ell \times n}$ with $\ell \leq M$ and a row labeling function $\rho: [\ell] \rightarrow A$, and a message $[m]_\tau \in \mathcal{M}_\lambda$, the encryption algorithm proceeds as follows:

- Sample an exponent $s \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and let $[c_1]_\tau = [s]_\tau + [m]_\tau$.
- Sample exponents $v_2, \dots, v_n, w_2, \dots, w_n \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and define $\mathbf{v} = (s, v_2, \dots, v_n)$ and $\mathbf{w} = (0, w_2, \dots, w_n)$.
- For each $x \in [\ell]$, sample an exponent $r_x \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and perform the following:
 - * Compute the following ciphertext components:

$$[c_{2,x}]_\tau = [\mathbf{m}_x^\top \mathbf{v}]_\tau + r_x [\alpha_{\rho(x)}]_\tau, \quad [c_{3,x}]_1 = [r_x]_1, \quad [c_{4,x}]_1 = r_x [y_{\rho(x)}]_1 + [\mathbf{m}_x^\top \mathbf{w}]_1,$$

where $\mathbf{m}_x^\top$ is the x^{th} row of $\mathbf{M}$.

- * Sample a set $S_{\text{ct},x} \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$ and let $[c_{5,x,i}]_1 = r_x [\gamma_{i,\rho(x)}]_1$ for $i \in S_{\text{ct},x}$ and $[c_{5,x,i}]_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1$ for $i \in [L] \setminus S_{\text{ct},x}$.

Output the ciphertext $\text{ct} = ((\mathbf{M}, \rho), [c_1]_\tau, \{([c_{2,x}]_\tau, [c_{3,x}]_1, [c_{4,x}]_1, \{[c_{5,x,i}]_1\}_{i \in [L]})\}_{x \in [\ell]})$.

- **Decrypt**($\{\text{sk}_{a,\text{gid}}\}_{a \in A}, \text{ct}$): On input a collection of secret keys $\{\text{sk}_{a,\text{gid}}\}_{a \in A}$ for a (fixed) global identifier $\text{gid} \in \mathcal{G}_\lambda$ and a set of attributes A where each $\text{sk}_{a,\text{gid}} = ([\delta_{a,\text{gid}}]_2, \{[\beta_{i,a,\text{gid}}]_2\}_{i \in S_{a,\text{gid}}})$, together with an input ciphertext $\text{ct} = ((\mathbf{M}, \rho), [c_1]_\tau, \{([c_{2,x}]_\tau, [c_{3,x}]_1, [c_{4,x}]_1, \{[c_{5,x,i}]_1\}_{i \in [L]})\}_{x \in [\ell]})$, the decryption algorithm proceeds as follows:
 - If A does not satisfy the policy $(\mathbf{M}, \rho)$, then the decryption algorithm outputs $\perp$.
 - Otherwise, let $X = \{x \in [\ell] : \rho(x) \in A\}$ be the indices of the rows of $\mathbf{M}$ associated with the attributes in A and write $X = \{x_1, \dots, x_{|X|}\}$. Now, let $\mathbf{M}_X$ be the matrix formed by taking the subset of rows in $\mathbf{M}$ indexed by X . Since A is authorized, there exists a vector $\omega_A = (\omega_{A,1}, \dots, \omega_{A,|X|}) \in \mathbb{Z}_p^{|X|}$ such that $\omega_A^\top \mathbf{M}_X = \mathbf{e}_1^\top$.
 - Compute $[h_{\text{gid}}]_2 \leftarrow H(\text{gid})$.
 - Output the following message:

$$[m]_\tau = [c_1]_\tau - \sum_{i \in [|X|]} \omega_{A,i} \cdot \left([c_{2,x_i}]_\tau - [c_{3,x_i}]_1 \cdot [\delta_{\rho(x_i),\text{gid}}]_2 + [c_{4,x_i}]_1 \cdot [h_{\text{gid}}]_2 + \sum_{j \in S_{\rho(x_i),\text{gid}}} [c_{5,x_i,j}]_1 \cdot [\beta_{j,\rho(x_i),\text{gid}}]_2 \right).$$

Theorem 4.8 (Correctness). *Construction 4.7 is statistically correct with error $\varepsilon(\lambda) = M \cdot L_{\text{sk}}(1 - L_{\text{ct}}/L)$.*

Proof. Take any $\lambda, M \in \mathbb{N}$ and any pp in the support of $\text{GlobalSetup}(1^\lambda, 1^M)$ where pp = ($\mathcal{G}, H$). Now take any set of attributes $A \subseteq \mathcal{U}_\lambda$ and, for each $a \in A$, take any $(\text{mpk}_a, \text{msk}_a)$ in the support of $\text{AuthSetup}(\text{pp}, a)$ where $\text{mpk}_a = ([\alpha_a]_\tau, [y_a]_1, \{[\gamma_{i,a}]_1\}_{i \in [L]})$ and $\text{msk}_a = (\alpha_a, y_a, \{\gamma_{i,a}\}_{i \in [L]})$. Next, take any policy $(\mathbf{M}, \rho)$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times n}$ for

$\ell \leq M$, and $\rho: [\ell] \rightarrow A$, and take any $A' \subseteq A$ satisfying $(\mathbf{M}, \rho)$. Furthermore, take any $\text{gid} \in \mathcal{G}_\lambda$ and, for all $a \in A'$, any $\text{sk}_{a,\text{gid}}$ in the support of $\text{KeyGen}(\text{msk}_a, \text{gid})$. We have that each secret key $\text{sk}_{a,\text{gid}} = ([\delta_{a,\text{gid}}]_2, \{[\beta_{i,a,\text{gid}}]_2\}_{i \in S_{a,\text{gid}}})$ where

$$[\delta_{a,\text{gid}}]_2 = \left[\alpha_a + y_a h_{\text{gid}} + \sum_{i \in S_{a,\text{gid}}} \gamma_{i,a} \beta_{i,a,\text{gid}} \right]_2.$$

Finally, for any message $[m]_{\text{T}} \in \mathcal{M}_\lambda$, let $\text{ct} \leftarrow \text{Encrypt}(\{\text{mpk}_a\}_{a \in A}, (\mathbf{M}, \rho), [m]_{\text{T}})$ where $\text{ct} = ((\mathbf{M}, \rho), [c_1]_{\text{T}}, \{([c_{2,x}]_{\text{T}}, [c_{3,x}]_1, [c_{4,x}]_1, \{[c_{5,x,i}]_1\}_{i \in [L]})\}_{x \in [\ell]})$ for

$$\begin{aligned} [c_{2,x}]_{\text{T}} &= [\mathbf{m}_x^{\text{T}} \mathbf{v}]_{\text{T}} + r_x [\alpha_{\rho(x)}]_{\text{T}} = [\mathbf{m}_x^{\text{T}} \mathbf{v} + r_x \alpha_{\rho(x)}]_{\text{T}}, & [c_1]_{\text{T}} &= [s]_{\text{T}} + [m]_{\text{T}} = [s + m]_{\text{T}}, \\ [c_{3,x}]_1 &= [r_x]_1, \\ [c_{4,x}]_1 &= r_x [y_{\rho(x)}]_1 + [\mathbf{m}_x^{\text{T}} \mathbf{w}]_1 = [r_x y_{\rho(x)} + \mathbf{m}_x^{\text{T}} \mathbf{w}]_1, & [c_{5,x,i}]_1 &= \begin{cases} [r_x \gamma_{i,\rho(x)}]_1 & i \in S_{\text{ct},x} \\ [r'_{x,i}]_1 & i \notin S_{\text{ct},x} \end{cases}, \end{aligned}$$

where $\mathbf{m}_x^{\text{T}}$ is the x^{th} row of $\mathbf{M}$ and, for each $x \in [\ell]$ and $i \notin S_{\text{ct},x}$, $r'_{x,i} \in \mathbb{Z}_p$ denotes the discrete logarithm of the uniformly random group element $[c_{5,x,i}]_1 \stackrel{\mathcal{R}}{\leftarrow} \mathbb{G}_1$ sampled by Encrypt .

Now consider $[m^*]_{\text{T}} \leftarrow \text{Decrypt}(\{\text{sk}_{a,\text{gid}}\}_{a \in A'}, \text{ct})$. As A' satisfies $(\mathbf{M}, \rho)$, there exists $\omega_{A'} = (\omega_{A',1}, \dots, \omega_{A',|X|}) \in \mathbb{Z}_p^{|X|}$ where $\omega_{A'}^{\text{T}} \mathbf{M}_X = \mathbf{e}_1^{\text{T}}$ and $X = \{x_1, \dots, x_{|X|}\} = \{x \in [\ell] : \rho(x) \in A'\}$ is the set of indices of the rows of $\mathbf{M}$ associated with the attributes in A' . Now, for each $i \in [|X|]$, let $a_i = \rho(x_i)$ and observe that the interpolation value

$$\begin{aligned} [z_i]_{\text{T}} &= [c_{2,x_i}]_{\text{T}} - [c_{3,x_i}]_1 \cdot [\delta_{a_i,\text{gid}}]_2 + [c_{4,x_i}]_1 \cdot [h_{\text{gid}}]_2 + \sum_{j \in S_{a_i,\text{gid}}} [c_{5,x_i,j}]_1 \cdot [\beta_{j,a_i,\text{gid}}]_2 \\ &= [\mathbf{m}_{x_i}^{\text{T}} \mathbf{v} + r_{x_i} \alpha_{a_i}]_{\text{T}} - \left[r_{x_i} \alpha_{a_i} + h_{\text{gid}} r_{x_i} y_{a_i} + r_{x_i} \sum_{j \in S_{a_i,\text{gid}}} \gamma_{j,a_i} \beta_{j,a_i,\text{gid}} \right]_{\text{T}} + [h_{\text{gid}} r_{x_i} y_{a_i} + h_{\text{gid}} \mathbf{m}_{x_i}^{\text{T}} \mathbf{w}]_{\text{T}} \\ &\quad + \sum_{j \in S_{a_i,\text{gid}} \cap S_{\text{ct},x_i}} [r_{x_i} \gamma_{j,a_i} \beta_{j,a_i,\text{gid}}]_{\text{T}} + \sum_{j \in S_{a_i,\text{gid}} \setminus S_{\text{ct},x_i}} [r'_{x_i,j} \beta_{j,a_i,\text{gid}}]_{\text{T}} \\ &= \left[\mathbf{m}_{x_i}^{\text{T}} \mathbf{v} + h_{\text{gid}} \mathbf{m}_{x_i}^{\text{T}} \mathbf{w} + \sum_{j \in S_{a_i,\text{gid}} \setminus S_{\text{ct},x_i}} (r'_{x_i,j} - r_{x_i} \gamma_{j,a_i}) \beta_{j,a_i,\text{gid}} \right]_{\text{T}}. \end{aligned}$$

This means that

$$\begin{aligned} [m^*]_{\text{T}} &= [s + m]_{\text{T}} - \sum_{i \in [|X|]} \omega_{A',i} [z_i]_{\text{T}} \\ &= \left[s + m - \sum_{i \in [|X|]} \omega_{A',i} \cdot \left(\mathbf{m}_{x_i}^{\text{T}} \mathbf{v} + h_{\text{gid}} \mathbf{m}_{x_i}^{\text{T}} \mathbf{w} + \sum_{j \in S_{a_i,\text{gid}} \setminus S_{\text{ct},x_i}} (r'_{x_i,j} - r_{x_i} \gamma_{j,a_i}) \beta_{j,a_i,\text{gid}} \right) \right]_{\text{T}} \\ &= \left[s + m - \omega_{A'}^{\text{T}} \mathbf{M}_X \mathbf{v} - h_{\text{gid}} \cdot \omega_{A'}^{\text{T}} \mathbf{M}_X \mathbf{w} - \sum_{i \in [|X|]} \omega_{A',i} \cdot \left(\sum_{j \in S_{a_i,\text{gid}} \setminus S_{\text{ct},x_i}} (r'_{x_i,j} - r_{x_i} \gamma_{j,a_i}) \beta_{j,a_i,\text{gid}} \right) \right]_{\text{T}} \\ &= \left[s + m - \mathbf{e}_1^{\text{T}} \mathbf{v} - h_{\text{gid}} \cdot \mathbf{e}_1^{\text{T}} \mathbf{w} - \sum_{i \in [|X|]} \omega_{A',i} \cdot \left(\sum_{j \in S_{a_i,\text{gid}} \setminus S_{\text{ct},x_i}} (r'_{x_i,j} - r_{x_i} \gamma_{j,a_i}) \beta_{j,a_i,\text{gid}} \right) \right]_{\text{T}} \\ &= \left[s + m - s - \sum_{i \in [|X|]} \omega_{A',i} \cdot \left(\sum_{j \in S_{a_i,\text{gid}} \setminus S_{\text{ct},x_i}} (r'_{x_i,j} - r_{x_i} \gamma_{j,a_i}) \beta_{j,a_i,\text{gid}} \right) \right]_{\text{T}} \\ &= \left[m - \sum_{i \in [|X|]} \omega_{A',i} \cdot \left(\sum_{j \in S_{a_i,\text{gid}} \setminus S_{\text{ct},x_i}} (r'_{x_i,j} - r_{x_i} \gamma_{j,a_i}) \beta_{j,a_i,\text{gid}} \right) \right]_{\text{T}}. \end{aligned}$$

Observe that if $S_{a_i, \text{gid}} \subseteq S_{\text{ct}, x_i}$ for all $i \in [|X|]$, then $S_{a_i, \text{gid}} \setminus S_{\text{ct}, x_i} = \emptyset$ for all $i \in [|X|]$ and consequently the blinding term vanishes, so $[m^*]_{\text{T}} = [m]_{\text{T}}$ and decryption succeeds. It remains to bound the probability of decryption failure. For any given $i \in [|X|]$, we have that

$$\Pr[S_{a_i, \text{gid}} \not\subseteq S_{\text{ct}, x_i}] \leq \sum_{j \in S_{a_i, \text{gid}}} \Pr[j \notin S_{\text{ct}, x_i}] = L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L}\right).$$

By a union bound and the fact that $\ell \leq M$, the decryption error is

$$\begin{aligned} \Pr[[m^*]_{\text{T}} \neq [m]_{\text{T}}] &\leq \Pr[\exists i \in [|X|] : S_{a_i, \text{gid}} \not\subseteq S_{\text{ct}, x_i}] \\ &\leq \ell \cdot L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L}\right) \leq M \cdot L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L}\right). \end{aligned} \quad \square$$

Theorem 4.9 (Security). *Suppose $2^\lambda < p = 2^{\Theta(\lambda)}$, $L = \text{poly}_1(\lambda, M)$ and $K = \text{poly}_2(\lambda, M)$, $L_{\text{sk}} = \omega(\log \lambda)$ and $K \cdot L_{\text{sk}} \geq \kappa \cdot M \cdot L$ for some constant $\kappa \geq 2$, and $L - L_{\text{ct}} = \omega(\log \lambda)$. Then [Construction 4.7](#) is unique-attribute-identity semantically secure for the subset of policies $\mathcal{P}' = \{\mathcal{P}'_{M, \lambda}\}_{M, \lambda \in \mathbb{N}}$ where $\mathcal{P}'_{M, \lambda} \subseteq \mathcal{P}_{M, \lambda}$ consists of policies that achieve linear independence for unauthorized sets in the generic bilinear group model of order p .*

Proof. Consider the following hybrids parameterized by a bit b and a security parameter λ and, implicitly, an efficient admissible adversary $\mathcal{A}$. We use [green](#) to denote changes between hybrids.

- **Hybrid $\text{Hyb}_0^{(b)}(\lambda)$:** This hybrid is equivalent to an admissible adversary $\mathcal{A}$ playing the unique-attribute-identity semantic security game against [Construction 4.7](#) where the underlying bilinear group is modeled as a *generic* bilinear group per [Definition 2.2](#). We let q_{AS} , q_{KG} , q_{BGGM} , and q_{ZT} denote upper bounds on the respective number of authority setup, key generation, generic bilinear group operation (i.e., encode, add, and pair oracles), and zero-test oracle queries $\mathcal{A}$ makes. We additionally let $N = N(\lambda)$ denote an upper bound on the size of the number of columns n in the challenge policy $P^* = (\mathbf{M}^*, \rho^*)$ where $\mathbf{M}^* \in \mathbb{Z}_p^{\ell \times n}$. The hybrid proceeds as follows:

- **Setup phase:** The challenger, upon receiving a policy-size bound 1^M from the adversary, performs the setup procedure as follows:

- * The challenger samples a prime $2^\lambda < p = 2^{\Theta(\lambda)}$, sets the BGGM handle space $\mathcal{H} = \{0, 1\}^\lambda$, and also initializes the table $\mathcal{T}$ which maps handles to exponents.
- * Next, it samples exponents $u_1, \dots, u_K \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and executes $\text{BGGM.Encode}(u_i, 2)$ to produce a handle $\langle u_i \rangle$ for each $i \in [K]$. Define the hash function $H'(X) := \sum_{i \in [K]} X^{i-1} u_i$.

The challenger sends the parameters $\text{pp} = (p, \{\langle u_i \rangle\}_{i \in [K]})$ to the adversary $\mathcal{A}$.

- **Pre-challenge query phase:** The adversary $\mathcal{A}$ may make authority setup, authority corruption, and key-generation queries of its choosing (subject to admissibility) which proceed as follows:

- * **Authority-setup query:** On the ζ^{th} authority-setup query on an attribute $a_\zeta \in \mathcal{U}_\lambda$, the challenger samples exponents $\alpha_\zeta, y_\zeta, \gamma_{1, \zeta}, \dots, \gamma_{L, \zeta} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$. It then executes $\text{BGGM.Encode}(\alpha_\zeta, \text{T})$ to produce $\langle \alpha_\zeta \rangle$, $\text{BGGM.Encode}(y_\zeta, 1)$ to produce $\langle y_\zeta \rangle$, and $\text{BGGM.Encode}(\gamma_{i, \zeta}, 1)$ to produce $\langle \gamma_{i, \zeta} \rangle$ for each $i \in [L]$. It sends the authority public key $\text{mpk}_\zeta = (\langle \alpha_\zeta \rangle, \langle y_\zeta \rangle, \{\langle \gamma_{i, \zeta} \rangle\}_{i \in [L]})$ to the adversary $\mathcal{A}$.
- * **Authority-corruption query:** On a corruption query on attribute $a_\zeta \in \mathcal{U}_\lambda$, the challenger sends the authority secret key $\text{msk}_\zeta = (\alpha_\zeta, y_\zeta, \{\gamma_{i, \zeta}\}_{i \in [L]})$ to the adversary $\mathcal{A}$.
- * **Key-generation query:** On the ξ^{th} key-generation query on an attribute $a_\xi \in \mathcal{U}_\lambda$ and a global identifier $\text{gid}_\xi \in \mathcal{G}_\lambda$ (where the pair (a_ξ, gid_ξ) is unique, as we are in the unique-attribute-identity setting), the challenger generates a secret key sk_ξ as follows:
 - The challenger samples a set $S_\xi \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$.
 - Next, for each $i \in S_\xi$, it samples an exponent $\beta_{i, \xi} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and executes $\text{BGGM.Encode}(\beta_{i, \xi}, 2)$ which produces a handle $\langle \beta_{i, \xi} \rangle$.

- Finally, it computes $\delta_\xi = \alpha_\xi + y_\xi h_\xi + \sum_{i \in S_\xi} \gamma_{i,\xi} \beta_{i,\xi}$ where $h_\xi \leftarrow H'(\text{gid}_\xi)$ and where α_ξ , y_ξ , and each $\gamma_{i,\xi}$ are from the authority secret key msk_ξ where $\xi = \zeta(\xi)$ is the index of the authority setup query on a_ξ . It then executes $\text{BGM.Encode}(\delta_\xi, 2)$ to produce $\langle \delta_\xi \rangle$.

The challenger sends the secret key $\text{sk}_\xi = (\langle \delta_\xi \rangle, \{\langle \beta_{i,\xi} \rangle\}_{i \in S_\xi})$ to the adversary $\mathcal{A}$.

During this phase, $\mathcal{A}$ may also query the generic group oracles. We reiterate their functionality below:

- * $\text{BGM.Encode}(x, v)$: On input an exponent $x \in \mathbb{Z}_p$ and an index $v \in \{1, 2, T\}$, the oracle samples a handle $\langle h \rangle \xleftarrow{\mathcal{R}} \mathcal{H}$ and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
 - * $\text{BGM.Add}(\langle h_1 \rangle, \langle h_2 \rangle)$: On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if $v_1 \neq v_2$. It then computes $x = x_1 + x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{\mathcal{R}} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v_1)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
 - * $\text{BGM.Pair}(\langle h_1 \rangle, \langle h_2 \rangle)$: On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if either $v_1 \neq 1$ or $v_2 \neq 2$. It then computes $x = x_1 x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{\mathcal{R}} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, T)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
 - * $\text{BGM.ZeroTest}(\langle h \rangle)$: On input a handle $\langle h \rangle \in \mathcal{H}$, the oracle retrieves $\mathcal{T}(\langle h \rangle) = (x, v)$ and aborts with output “invalid” if the entry is undefined. It returns “zero” if $x = 0 \in \mathbb{Z}_p$ and “non-zero” otherwise.
- **Challenge phase:** In this phase, the adversary $\mathcal{A}$ outputs an admissible challenge policy $P^* = (\mathbf{M}^*, \rho^*)$ over a set of attributes $A^* \subseteq \mathcal{U}_\lambda$ (where $\mathbf{M}^* \in \mathbb{Z}_p^{\ell \times n}$ with $\ell \leq M$ and $\rho^*: [\ell] \rightarrow A^*$), together with handles $\langle m_0^* \rangle, \langle m_1^* \rangle$ for $\mathcal{A}$ ’s challenge messages such that $\mathcal{T}(\langle m_0^* \rangle) = (m_0^*, T)$ and $\mathcal{T}(\langle m_1^* \rangle) = (m_1^*, T)$. Namely, both messages are in the target group. For each corrupt attribute $a^* \in A^* \setminus A$ (where A denotes the set of attributes queried to the authority-setup oracle at this point), $\mathcal{A}$ also provides authority public-key handles $\text{mpk}_{a^*} = (\langle \alpha_{a^*} \rangle, \langle y_{a^*} \rangle, \{\langle \gamma_{i,a^*} \rangle\}_{i \in [L]})$ such that $\mathcal{T}(\langle \alpha_{a^*} \rangle) = (\alpha_{a^*}, T)$ and $\mathcal{T}(\langle y_{a^*} \rangle) = (y_{a^*}, 1)$, and $\mathcal{T}(\langle \gamma_{i,a^*} \rangle) = (\gamma_{i,a^*}, 1)$ for each $i \in [L]$. Namely, the authority public-key components are in the correct groups. The challenger produces the challenge ciphertext ct^* encrypting $\langle m_b^* \rangle$ under P^* as follows:
- * The challenger samples an exponent $s \xleftarrow{\mathcal{R}} \mathbb{Z}_p$, retrieves the entry $\mathcal{T}(\langle m_b^* \rangle) = (m_b^*, T)$, and then runs $\text{BGM.Encode}(s + m_b^*, T)$ to produce a handle $\langle c_1^* \rangle$.
 - * Next, it samples exponents $v_2, \dots, v_n, w_2, \dots, w_n \xleftarrow{\mathcal{R}} \mathbb{Z}_p$ and then defines the vectors $\mathbf{v} = (s, v_2, \dots, v_n)$ and $\mathbf{w} = (0, w_2, \dots, w_n)$.
 - * For each $x \in [\ell]$, let $a_x = \rho^*(x)$, and let $\mathbf{m}_x^{*\top}$ denote the x^{th} row of $\mathbf{M}^*$. If $a_x \in A$, let ζ_x be the index of the authority-setup query $\mathcal{A}$ made with a_x as input. The challenger samples an exponent $r_x \xleftarrow{\mathcal{R}} \mathbb{Z}_p$ and does the following:
 - The challenger executes $\text{BGM.Encode}(\mathbf{m}_x^{*\top} \mathbf{v} + r_x \alpha_{\zeta_x}, T)$ to produce $\langle c_{2,x}^* \rangle$.
 - Next, it executes $\text{BGM.Encode}(r_x, 1)$ to produce $\langle c_{3,x}^* \rangle$.
 - Next, it executes $\text{BGM.Encode}(\mathbf{m}_x^{*\top} \mathbf{w} + r_x y_{\zeta_x}, 1)$ to produce $\langle c_{4,x}^* \rangle$.
 - Finally, it samples a set $S_x^* \xleftarrow{\mathcal{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. For each $i \in [L]$, it computes $c_{5,x,i}^* = r_x \gamma_{i,\zeta_x}$ if $i \in S_x^*$ and samples $c_{5,x,i}^* \xleftarrow{\mathcal{R}} \mathbb{Z}_p$ otherwise, then runs $\text{BGM.Encode}(c_{5,x,i}^*, 1)$ to produce $\langle c_{5,x,i}^* \rangle$.
- If $a_x \notin A$, the challenger samples an exponent $r_x \xleftarrow{\mathcal{R}} \mathbb{Z}_p$ and does the following:
- The challenger retrieves the entry $\mathcal{T}(\langle \alpha_{a_x} \rangle) = (\alpha_{a_x}, T)$, and then executes $\text{BGM.Encode}(\mathbf{m}_x^{*\top} \mathbf{v} + r_x \alpha_{a_x}, T)$ to produce $\langle c_{2,x}^* \rangle$.
 - Next, it executes $\text{BGM.Encode}(r_x, 1)$ to produce $\langle c_{3,x}^* \rangle$.
 - Next, it retrieves the entry $\mathcal{T}(\langle y_{a_x} \rangle) = (y_{a_x}, 1)$, then executes $\text{BGM.Encode}(\mathbf{m}_x^{*\top} \mathbf{w} + r_x y_{a_x}, 1)$ to produce $\langle c_{4,x}^* \rangle$.
 - Finally, it samples a set $S_x^* \xleftarrow{\mathcal{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. For each $i \in [L]$, it does the following. If $i \in S_x^*$, it retrieves the entry $\mathcal{T}(\langle \gamma_{i,a_x} \rangle) = (\gamma_{i,a_x}, 1)$, then executes $\text{BGM.Encode}(r_x \gamma_{i,a_x}, 1)$ to produce $\langle c_{5,x,i}^* \rangle$. Otherwise, it samples $c_{5,x,i}^* \xleftarrow{\mathcal{R}} \mathbb{Z}_p$ and executes $\text{BGM.Encode}(c_{5,x,i}^*, 1)$ to produce $\langle c_{5,x,i}^* \rangle$.

The challenger sends the challenge ciphertext

$$\text{ct}^* = ((\mathbf{M}^*, \rho^*), \langle c_1^* \rangle, \{(\langle c_{2,x}^* \rangle, \langle c_{3,x}^* \rangle, \langle c_{4,x}^* \rangle, \{\langle c_{5,x,i}^* \rangle\}_{i \in [L]})\}_{x \in [\ell]}),$$

to the adversary $\mathcal{A}$.

- **Post-challenge query phase:** The adversary $\mathcal{A}$ may again make authority setup, authority corruption, and key-generation queries subject to admissibility. Namely, for every global identifier $\text{gid} \in \mathcal{G}_\lambda$, the corrupted attributes (including those provided by the adversary during the challenge phase) together with the set A_{gid} of attributes queried to the key-generation oracle for gid do not satisfy the challenge policy P^* , and no authority-setup query is made on an attribute $a \in \mathcal{U}_\lambda$ that was previously queried to the authority-setup oracle or supplied by the adversary during the challenge phase and all corruption and key-generation queries are made on attributes previously set up. Furthermore, recall that we are in the unique-attribute-identity setting, and thus $\mathcal{A}$ may only make key-generation queries on distinct (a_ξ, gid_ξ) pairs. The challenger responds using the same procedures as the pre-challenge query phase. During this phase, $\mathcal{A}$ may again make encode, add, pair, and zero-test oracle queries as specified above.
- **Output phase:** The adversary outputs a bit b^* which is the output of the game.
- **Hybrid $\text{Hyb}_1^{(b)}(\lambda)$:** This hybrid is equivalent to hybrid $\text{Hyb}_0^{(b)}$, except we add the following halting condition:
 - **Key-generation queries:** Let $K^* = K/M$. If there exists a collection of K^* sets $S_{\xi_1}, \dots, S_{\xi_{K^*}} \subseteq [L]$ sampled during distinct key-generation queries (i.e., $\xi_1, \dots, \xi_{K^*}$ are all different) on the same attribute a such that $|\bigcup_{i \in [K^*]} S_{\xi_i}| \leq L/\kappa$, halt with output 0.
- **Hybrid $\text{Hyb}_2^{(b)}(\lambda)$:** This hybrid is equivalent to hybrid $\text{Hyb}_1^{(b)}$, except we add the following halting condition:
 - **Generic group oracle queries:** If BGGM.Encode , BGGM.Add , or BGGM.Pair returns “collision” because the sampled handle $\langle h \rangle$ has previously been sampled (i.e., $\mathcal{T}(\langle h \rangle)$ is already defined), halt with output 0.
- **Hybrid $\text{Hyb}_3^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_2^{(b)}$ (including its handle collision-abort rule) except that it augments the challenger so that, in addition to maintaining handles that refer to group elements, it also keeps track of two kinds of formal polynomials to which the handles map. Specifically, handles given to the adversary map to polynomials over a set of formal variables $\hat{\mathcal{V}}$, while the challenger simultaneously maintains polynomials over a set of formal variables $\tilde{\mathcal{V}}$ that it will use to evaluate zero-test queries. We refer to the former as the *adversary’s* formal polynomials and the latter as the *challenger’s* formal polynomials. To distinguish the representations used below, a plain symbol x denotes an actual exponent in $\mathbb{Z}_p$, $[x]_v$ denotes the corresponding concrete group element in group $\mathbb{G}_v$, and $\langle x \rangle$ denotes an opaque handle for that group element in the generic-group game. A hatted polynomial $\hat{p}$ records, using only the components exposed to the adversary, the formal algebraic expression associated with a handle, while a tilded polynomial $\tilde{p}$ expands the same handle in terms of the challenger’s underlying sampled values.

The set of variables $\hat{\mathcal{V}}$ over which the adversary’s formal polynomials are defined captures the components the construction returns during global setup, authority setup, key generation, and encryption. In detail, $\hat{\mathcal{V}}$ is specified as follows:

$$\begin{aligned} \hat{\mathcal{V}}_1 &= \{\hat{y}_\zeta\}_{\zeta \in [q_{\text{AS}}]} \cup \{\hat{y}_{i,\zeta}\}_{(i,\zeta) \in [L] \times [q_{\text{AS}}]} \cup \{\hat{c}_{3,x}^*\}_{x \in [M]} \cup \{\hat{c}_{4,x}^*\}_{x \in [M]} \cup \{\hat{c}_{5,x,i}^*\}_{(x,i) \in [M] \times [L]}, \\ \hat{\mathcal{V}}_2 &= \{\hat{u}_j\}_{j \in [K]} \cup \{\hat{\delta}_\xi\}_{\xi \in [q_{\text{KG}}]} \cup \{\hat{\rho}_{i,\xi}\}_{(i,\xi) \in [L] \times [q_{\text{KG}}]}, \\ \hat{\mathcal{V}}_T &= \{\hat{c}_1^*\} \cup \{\hat{\alpha}_\zeta\}_{\zeta \in [q_{\text{AS}}]} \cup \{\hat{c}_{2,x}^*\}_{x \in [M]}, \\ \hat{\mathcal{V}} &= \hat{\mathcal{V}}_1 \cup \hat{\mathcal{V}}_2 \cup \hat{\mathcal{V}}_T. \end{aligned}$$

The set of variables $\tilde{\mathcal{V}}$ over which the challenger’s formal polynomials are defined captures the group elements sampled when running global setup, authority setup, key generation, and encryption. In detail, $\tilde{\mathcal{V}}$ is specified

as follows:

$$\begin{aligned}\tilde{\mathcal{V}} = \{ \tilde{s} \} \cup \{ \tilde{\alpha}_\zeta, \tilde{y}_\zeta \}_{\zeta \in [q_{AS}]} \cup \{ \tilde{y}_{i,\zeta} \}_{(i,\zeta) \in [L] \times [q_{AS}]} \cup \{ \tilde{w}_k, \tilde{v}_k \}_{k \in [2,N]} \cup \{ \tilde{u}_j \}_{j \in [K]} \\ \cup \{ \tilde{r}_x \}_{x \in [M]} \cup \{ \tilde{r}'_{x,i} \}_{(x,i) \in [M] \times [L]} \cup \{ \tilde{\beta}_{i,\xi} \}_{(i,\xi) \in [L] \times [q_{KG}]}.\end{aligned}$$

At any time t in the hybrid, define an assignment $\phi_t: \tilde{\mathcal{V}} \rightarrow \mathbb{Z}_p$ that maps each challenger variable $\tilde{v} \in \tilde{\mathcal{V}}$ to the corresponding value $v \in \mathbb{Z}_p$ if it has been sampled by time t and to 0 otherwise. We let $\text{Eval}_{\phi_t}: \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p$ denote the standard polynomial-evaluation ring homomorphism induced by ϕ_t . Thus, in particular, the variables $\tilde{r}'_{x,i}$ and $\tilde{\beta}_{i,\xi}$ evaluate to 0 until the corresponding values have been sampled.

With adversary and challenger formal polynomials defined, the hybrid proceeds as follows:

- **Setup phase:** The challenger performs the setup procedure with the following additional steps:
 - * When the challenger initializes the table $\mathcal{T}$, it additionally initializes tables $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$, along with an empty set Z of authority-setup query indices for corrupted authorities.
 - * With handles $\langle u_i \rangle$ for each $i \in [K]$, it adds the entries $\langle u_i \rangle \mapsto \hat{u}_i$ to $\hat{\mathcal{T}}$ and $\langle u_i \rangle \mapsto \tilde{u}_i$ to $\tilde{\mathcal{T}}$.
- **Pre-challenge query phase:** For each authority setup, authority corruption, and key-generation query, the challenger responds with the following additional steps:
 - * **Authority-setup query:** For the ζ^{th} authority-setup query on attribute a_ζ , with the handles $\langle \alpha_\zeta \rangle$, $\langle y_\zeta \rangle$, and $\langle y_{i,\zeta} \rangle$ for each $i \in [L]$, the challenger additionally adds the entries $\langle \alpha_\zeta \rangle \mapsto \hat{\alpha}_\zeta$, $\langle y_\zeta \rangle \mapsto \hat{y}_\zeta$, and $\langle y_{i,\zeta} \rangle \mapsto \hat{y}_{i,\zeta}$ to $\hat{\mathcal{T}}$ and $\langle \alpha_\zeta \rangle \mapsto \tilde{\alpha}_\zeta$, $\langle y_\zeta \rangle \mapsto \tilde{y}_\zeta$, and $\langle y_{i,\zeta} \rangle \mapsto \tilde{y}_{i,\zeta}$ to $\tilde{\mathcal{T}}$.
 - * **Authority-corruption query:** On a corruption query on attribute a_ζ , the challenger defines the ring homomorphism $\text{Corr}_\zeta: \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ induced by

$$\begin{aligned}\tilde{\alpha}_\zeta &\mapsto \alpha_\zeta, \\ \tilde{y}_\zeta &\mapsto y_\zeta, & \text{and } \tilde{v} &\mapsto \tilde{v} \text{ otherwise,} \\ \tilde{y}_{i,\zeta} &\mapsto y_{i,\zeta} \text{ for } i \in [L]\end{aligned}$$

and adds ζ to Z . Label the elements of $Z = \{\zeta'_1, \dots, \zeta'_m\}$ and define $\text{Corr}_Z = \text{Corr}_{\zeta'_1} \circ \dots \circ \text{Corr}_{\zeta'_m}$ (notably, this is the identity mapping when Z is empty). For each handle $\langle h \rangle$ in $\tilde{\mathcal{T}}$, let $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ be the challenger's formal polynomial indexed by $\langle h \rangle$ and replace the entry $\langle h \rangle \mapsto \tilde{h}$ with the new entry $\langle h \rangle \mapsto \text{Corr}_Z(\tilde{h})$. Note that, after corruption, the variables of a corrupt authority do not appear in $\tilde{\mathcal{T}}$. Furthermore, because ϕ_t maps the variables $\tilde{\alpha}_\zeta$, $\tilde{y}_\zeta$, and $\tilde{y}_{i,\zeta}$ of each corrupt authority $\zeta \in Z$ to α_ζ , y_ζ , and $y_{i,\zeta}$ respectively (i.e., the same values Corr_Z itself substitutes), we have that

$$\text{Eval}_{\phi_t}(\text{Corr}_Z(\tilde{h})) = \text{Eval}_{\phi_t}(\tilde{h}).$$

- * **Key-generation query:** For the ξ^{th} key-generation query on attribute a_ξ and global identifier gid_ξ , the challenger additionally does the following:
 - With the handle $\langle \beta_{i,\xi} \rangle$ for each $i \in S_\xi$, it adds the entry $\langle \beta_{i,\xi} \rangle \mapsto \hat{\beta}_{i,\xi}$ to $\hat{\mathcal{T}}$ and $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ to $\tilde{\mathcal{T}}$.
 - When it computes $\delta_\xi = \alpha_\zeta + y_\zeta h_\xi + \sum_{i \in S_\xi} y_{i,\zeta} \beta_{i,\xi}$ (where α_ζ , y_ζ , and each $y_{i,\zeta}$ are from the authority secret key msk_ζ where $\zeta = \zeta(\xi)$ is the index of the authority setup query on a_ξ .) to produce the handle $\langle \delta_\xi \rangle$, it also adds the entries $\langle \delta_\xi \rangle \mapsto \hat{\delta}_\xi$ to $\hat{\mathcal{T}}$ and $\langle \delta_\xi \rangle \mapsto \text{Corr}_Z(\tilde{\alpha}_\zeta + \tilde{y}_\zeta \cdot \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\zeta} \tilde{\beta}_{i,\xi})$ to $\tilde{\mathcal{T}}$. Recall that $h_\xi = \sum_{j \in [K]} \text{gid}_\xi^{j-1} u_j$.

When $\mathcal{A}$ makes generic group oracle queries, the challenger performs the following additional steps:

- * **BGGM.Encode(x, v):** In addition to adding the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger adds the entry $\langle h \rangle \mapsto x$ to both $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ where x denotes the constant polynomial.

- * $\text{BGGM.Add}(\langle h_1 \rangle, \langle h_2 \rangle)$: When it adds the entry $\langle h \rangle \mapsto (x_1 + x_2, v_1)$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger additionally adds the entries $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$ and $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$, where both sums denote formal polynomial addition.
- * $\text{BGGM.Pair}(\langle h_1 \rangle, \langle h_2 \rangle)$: When it adds the entry $\langle h \rangle \mapsto (x_1 x_2, \mathsf{T})$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger additionally adds the entries $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$ and $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$, where both products denote multiplication in the formal polynomial ring.
- * $\text{BGGM.ZeroTest}(\langle h \rangle)$: If $\langle h \rangle$ is present in $\mathcal{T}$, then it returns “zero” if

$$\text{Eval}_{\phi_t}(\tilde{h}) = 0$$

for the challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ at the time t of the query, and returns “non-zero” otherwise.

– **Challenge phase:** This phase includes the following additional steps:

- * When the challenger produces the handle $\langle c_1^* \rangle$, it also adds the entries $\langle c_1^* \rangle \mapsto \hat{c}_1^*$ to $\hat{\mathcal{T}}$ and $\langle c_1^* \rangle \mapsto \tilde{s} + \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ to $\tilde{\mathcal{T}}$ where $\langle m_b^* \rangle$ is the handle for the challenge message.
- * Next, it defines vectors of challenger polynomial variables $\tilde{\mathbf{v}} := (\tilde{s}, \tilde{v}_2, \dots, \tilde{v}_n)$ and $\tilde{\mathbf{w}} := (0, \tilde{w}_2, \dots, \tilde{w}_n)$. Then, for each row $x \in [\ell]$ in the challenge policy $(\mathbf{M}^*, \rho^*)$, let $a_x = \rho^*(x)$ and let $\langle \alpha_{a_x} \rangle$, $\langle y_{a_x} \rangle$, and $\langle \gamma_{i,a_x} \rangle$ for $i \in [L]$ be the components of the authority public key mpk_{a_x} used to encrypt (i.e., the handles the challenger produced on the authority-setup query for a_x if $a_x \in A$ and the handles $\mathcal{A}$ supplied in this phase otherwise). In the former case, $\tilde{\mathcal{T}}(\langle \alpha_{a_x} \rangle) = \tilde{\alpha}_{\zeta_x}$ for the index ζ_x of that query (or the value α_{ζ_x} if a_x is corrupt), and similarly for $\langle y_{a_x} \rangle$ and $\langle \gamma_{i,a_x} \rangle$. The challenger then does the following:
 - With $\langle c_{2,x}^* \rangle$, it adds the entry $\langle c_{2,x}^* \rangle \mapsto \hat{c}_{2,x}^*$ to $\hat{\mathcal{T}}$ and $\langle c_{2,x}^* \rangle \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{v}} + \tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle \alpha_{a_x} \rangle))$ to $\tilde{\mathcal{T}}$.
 - With $\langle c_{3,x}^* \rangle$, it adds the entry $\langle c_{3,x}^* \rangle \mapsto \hat{c}_{3,x}^*$ to $\hat{\mathcal{T}}$ and $\langle c_{3,x}^* \rangle \mapsto \tilde{r}_x$ to $\tilde{\mathcal{T}}$.
 - With $\langle c_{4,x}^* \rangle$, it adds the entry $\langle c_{4,x}^* \rangle \mapsto \hat{c}_{4,x}^*$ to $\hat{\mathcal{T}}$ and $\langle c_{4,x}^* \rangle \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{w}} + \tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle y_{a_x} \rangle))$ to $\tilde{\mathcal{T}}$.
 - Finally, with $\langle c_{5,x,i}^* \rangle$ for each $i \in [L]$, it adds the entry $\langle c_{5,x,i}^* \rangle \mapsto \hat{c}_{5,x,i}^*$ to $\hat{\mathcal{T}}$. It additionally adds to $\tilde{\mathcal{T}}$ the entry $\langle c_{5,x,i}^* \rangle \mapsto \text{Corr}_Z(\tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle \gamma_{i,a_x} \rangle))$ if $i \in S_x^*$ and $\langle c_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$ otherwise.

– **Post-challenge query phase:** This phase is the same as the pre-challenge query phase. In particular, at the time t of each zero-test query, the challenger uses the current assignment ϕ_t , which now includes the values sampled during the challenge phase, and returns “zero” if $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ (where $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ is the challenger formal polynomial associated with the queried handle $\langle h \rangle$) and “non-zero” otherwise. Note that an authority-corruption query made in this phase will rewrite the entries added during the challenge phase.

• **Hybrid $\text{Hyb}_4^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_3^{(b)}$, except for the following change:

- **Zero-test oracle queries:** When the adversary $\mathcal{A}$ queries a handle $\langle h \rangle$ to the zero-test oracle, the challenger no longer evaluates the associated challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ under the current assignment ϕ_t . Instead, it returns “zero” if $\tilde{\mathcal{T}}(\langle h \rangle) \equiv 0$ in $\mathbb{Z}_p[\tilde{\mathcal{V}}]$ and “non-zero” otherwise. That is, this hybrid tests whether $\tilde{\mathcal{T}}(\langle h \rangle)$ is identically zero as a formal polynomial, while previous hybrids tested whether $\text{Eval}_{\phi_t}(\tilde{\mathcal{T}}(\langle h \rangle)) = 0$.

The behavior of the remaining hybrids is dependent on which authorities the adversary $\mathcal{A}$ has corrupted by a given zero-test query. We define $X_{\chi, \text{hon}}$ to be the set of rows $x \in [\ell]$ where $\rho^*(x) = a_\zeta$ for some $\zeta \notin Z_\chi$ (i.e., x is not for a corrupted attribute), where Z_χ denotes the set of authority-setup indices for corrupted authorities Z at the time of the χ^{th} post-challenge zero-test query. Furthermore, let $X_{\chi, \text{cor}} = [\ell] \setminus X_{\chi, \text{hon}}$. We let $X_{0, \text{hon}}$ and $X_{0, \text{cor}}$ denote these sets at the start of the post-challenge query phase. Note that, because $\mathcal{A}$ is admissible, it never makes an authority-setup query on attributes that were previously corrupted (either by a corruption query or by being supplied during the challenge phase).

We also define the following procedure given a set $X \subseteq [\ell]$. For each $x \in X$, the challenger samples a set $S_x^* \xleftarrow{\mathcal{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. Let $\text{SubS}_{X,Z} : \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ be the ring homomorphism induced by

$$\tilde{r}'_{x,i} \mapsto \text{Corr}_Z(\tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle \gamma_{i,a_x} \rangle)) \quad \text{for } x \in X, i \in S_x^* \quad \text{and} \quad \tilde{v} \mapsto \tilde{v} \quad \text{otherwise.}$$

We say that the challenger *samples and substitutes* (X, Z) if it replaces every entry $\langle h \rangle \mapsto \tilde{h}'$ of $\tilde{\mathcal{T}}$ with the entry $\langle h \rangle \mapsto \text{SubS}_{X,Z}(\tilde{h}')$ (i.e., the $\tilde{r}'_{x,i}$ components are substituted with $\text{Corr}_Z(\tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle y_{i,a_x} \rangle))$ for $x \in X$ and $i \in S_x^*$).

- **Hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_4^{(b)}$, except for the following change:
 - **Challenge phase:** This phase is modified so that for each $\langle c_{5,x,i}^* \rangle$ for $(x, i) \in X_{0,\text{hon}} \times [L]$, the challenger no longer samples S_x^* and instead always **adds to $\tilde{\mathcal{T}}$ the entry $\langle c_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$** . For each handle $\langle c_{5,x,i}^* \rangle$ for $(x, i) \in X_{0,\text{cor}} \times [L]$, it proceeds as before and samples a set $S_x^* \xleftarrow{R} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$, and then it adds the entry $\langle c_{5,x,i}^* \rangle \mapsto \text{Corr}_Z(\tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle y_{i,a_x} \rangle))$ to $\tilde{\mathcal{T}}$ if $i \in S_x^*$ and $\langle c_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$ to $\tilde{\mathcal{T}}$ otherwise. This process was previously done for *all* rows $x \in [\ell]$.
 - **Post-challenge query phase:** At the start of the post-challenge query phase, the challenger **additionally samples and substitutes for $(X_{0,\text{hon}}, Z)$** , where Z is the set of corrupted authorities *at this point*.
- **Hybrid $\text{Hyb}_{5,\chi}^{(b)}(\lambda)$ for $\chi \in [q_{\text{ZT}}]$:** This hybrid is equivalent to $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$, except for the following change:
 - **Post-challenge zero-test oracle queries:** On the χ^{th} post-challenge query to the zero-test oracle on a handle $\langle h \rangle$, the challenger defines the ring homomorphism $\text{Expd}_Z : \mathbb{Z}_p[\hat{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\hat{\mathcal{V}}]$ induced by

$$\hat{\delta}_\xi \mapsto \alpha_\xi + y_\xi \sum_{j \in [K]} \text{gid}_\xi^{j-1} \hat{u}_j + \sum_{i \in S_\xi} y_{i,\xi} \hat{\beta}_{i,\xi} \quad \text{for } \xi \in Z, \xi \in [q_{\text{KG}}] \text{ with } \xi = \zeta(\xi) \quad , \quad \hat{v} \mapsto \hat{v} \quad \text{otherwise}$$

i.e., Expd_Z expands the keys issued by corrupted authorities in terms of the known components that have been returned to $\mathcal{A}$. Then, it defines $\hat{h}_Z = \text{Expd}_{Z_\chi}(\hat{\mathcal{T}}(\langle h \rangle))$ and does the following:

- * **Message-dependent component:** If $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \neq 0$ (i.e., there is a $\hat{c}_1^*$ term in the expanded adversary formal polynomial associated with $\langle h \rangle$), **then the oracle returns “non-zero” (regardless of whether it is identical to the zero polynomial or not).**
- * **Uncorrupted-challenge-set-dependent component:** If $\text{coeff}_{\hat{c}_{5,x,i}^* \hat{v}}(\hat{h}_Z) \neq 0$ for some pair $(x, i) \in X_{\chi,\text{hon}} \times [L]$ and $\hat{v} \in \{1\} \cup \hat{\mathcal{V}}_2$, **then the oracle returns “non-zero” (regardless of whether it is identical to the zero polynomial or not).**
- * If neither of the cases above occur, and for the remaining post-challenge queries to the zero-test oracle, the challenger proceeds as in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$.

Note that because the hybrid is defined recursively, all post-challenge zero-test oracle queries before the χ^{th} query have already also been changed to include the above behavior.

- **Hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}(\lambda)$ for $\chi \in [q_{\text{ZT}}]$:** This hybrid is equivalent to $\text{Hyb}_{5,\chi}^{(b)}$, except for the following changes:
 - **Post-challenge zero-test oracle queries:** The challenger makes the following changes:
 - * **The challenger no longer samples and substitutes for $(X_{\chi-1,\text{hon}}, Z_{\chi-1})$ after the first $\chi - 1$ post-challenge queries to the zero-test oracle.**
 - * **Immediately before answering the χ^{th} post-challenge query to the zero-test oracle, the challenger samples and substitutes for $(X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}, Z_\chi)$ (i.e., for the rows whose authority was corrupted after the first $\chi - 1$ post-challenge queries to the zero-test oracle and before the χ^{th} one).**
 - * **Immediately after the χ^{th} post-challenge query to the zero-test oracle, the challenger samples and substitutes for $(X_{\chi,\text{hon}}, Z_\chi)$.**

Lemma 4.10. For $L_{\text{sk}} = \omega(\log \lambda)$ and $K \cdot L_{\text{sk}} \geq \kappa \cdot M \cdot L$ for some constant $\kappa \geq 2$, there exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,

$$|\Pr[\text{Hyb}_0^{(b)}(\lambda)] - \Pr[\text{Hyb}_1^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The hybrids are equivalent except for the event that the halting condition in $\text{Hyb}_1^{(b)}$ is triggered. Namely, letting $K^* = K/M$, the experiments are equivalent unless there exists some collection of sets $S_{\xi_1}, \dots, S_{\xi_{K^*}} \subseteq [L]$ sampled during (pre- or post-challenge) key-generation for K^* different keys $\xi_1, \dots, \xi_{K^*}$ for the same attribute a such that the union $|\bigcup_{i \in [K^*]} S_{\xi_i}| \leq L/\kappa$, or equivalently, unless there exists some set $T \subseteq [L]$ with $|T| = L/\kappa$ such that $S_{\xi_i} \subseteq T$ for all $i \in [K^*]$. We will bound the event that a collection of sets with union satisfying the above constraint is sampled on key-generation queries *even for different attributes*, which strictly contains the halting condition. Let bad denote this latter event.

We first consider the probability that a given set $S_{\xi_j} \xleftarrow{R} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$ is contained within a *fixed* set T of size L/κ . We have that

$$\Pr[S_{\xi_j} \subseteq T] \leq \binom{L/\kappa}{L_{\text{sk}}} \cdot \binom{L}{L_{\text{sk}}}^{-1} \leq \left(\frac{L/\kappa}{L}\right)^{L_{\text{sk}}} = \kappa^{-L_{\text{sk}}}.$$

Now, because each set $S_{\xi_1}, \dots, S_{\xi_{K^*}}$ is sampled independently of each other (and any queries the adversary does, as each key-generation query samples a fresh set), we have that

$$\Pr[\forall i \in [K^*] : S_{\xi_i} \subseteq T] = \Pr[S_{\xi_j} \subseteq T]^{K^*} \leq \kappa^{-K^* \cdot L_{\text{sk}}}.$$

To analyze the full bad event, observe that there are $\binom{q_{\text{KG}}}{K^*}$ possible collections of K^* sets and $\binom{L}{L/\kappa}$ possible target sets T . By applying the union bound and using the fact that $\kappa \geq 2$, we have that,

$$\begin{aligned} \Pr[\text{bad}] &\leq \binom{q_{\text{KG}}}{K^*} \cdot \binom{L}{L/\kappa} \cdot \Pr[\forall i \in [K^*] : S_{\xi_i} \subseteq T] \\ &\leq \left(\frac{e \cdot q_{\text{KG}}}{K^*}\right)^{K^*} \cdot \kappa^L \cdot \kappa^{-K^* \cdot L_{\text{sk}}} \\ &\leq \kappa^{K^* \cdot \log(e \cdot q_{\text{KG}}/K^*)/\log \kappa + L - K^* \cdot L_{\text{sk}}}. \end{aligned}$$

Finally, recall that $q_{\text{KG}} = \text{poly}(\lambda) \leq \lambda^{c_1}$ for all $\lambda \geq d_1$ for some $c_1, d_1 \in \mathbb{N}$, and that $K^* \cdot L_{\text{sk}} = K \cdot L_{\text{sk}}/M \geq \kappa \cdot L$. Consider any $c_2 > 0$. Because $L_{\text{sk}} = \omega(\log \lambda)$, there exists some $d_2 \in \mathbb{N}$ such that $L_{\text{sk}} \geq c_2 \log \lambda$ for all $\lambda \geq d_2$. Let $c = (1 - 1/\kappa) \log \kappa \cdot c_2 - c_1 - 1$ and $d = \max(d_1, d_2, 3)$. We have that,

$$\begin{aligned} \Pr[\text{bad}] &\leq \kappa^{K^* \cdot \log(e \cdot \lambda^{c_1})/\log \kappa + K^* \cdot L_{\text{sk}}/\kappa - K^* \cdot L_{\text{sk}}} && \text{for } \lambda \geq d_1 \\ &\leq \kappa^{K^* \cdot ((c_1+1) \log \lambda / \log \kappa - (1-1/\kappa) \cdot c_2 \log \lambda)} && \text{for } \lambda \geq d \\ &= \lambda^{-K^* \cdot c} && \text{for } \lambda \geq d \\ &\leq \lambda^{-c} && \text{for } \lambda \geq d, \text{ if } c \geq 0. \end{aligned}$$

Note that since $c_2 > 0$ was arbitrary, c can take any positive value. That is, for every $c > 0$, it holds that $\Pr[\text{bad}] \leq \lambda^{-c}$ for all sufficiently large λ , and so there exists some negligible function $\text{negl}(\cdot)$ such that

$$|\Pr[\text{Hyb}_0^{(b)}(\lambda)] - \Pr[\text{Hyb}_1^{(b)}(\lambda)]| \leq \Pr[\text{bad}] \leq \text{negl}(\lambda). \quad \square$$

Lemma 4.11. *For $L = \text{poly}_1(\lambda, M)$ and $K = \text{poly}_2(\lambda, M)$, there exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,*

$$|\Pr[\text{Hyb}_1^{(b)}(\lambda)] - \Pr[\text{Hyb}_2^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The two hybrids are identical except for the event that the halting condition in $\text{Hyb}_2^{(b)}$ is triggered (i.e., one of the handles sampled during the experiment is equal to a previously sampled handle). Let bad denote this event. Since each handle is sampled uniformly from $\mathcal{H} = \{0, 1\}^\lambda$, a union bound over pairs of sampled handles over the course of the hybrid (both sampled during adversarial queries and by the challenger) gives us

$$\Pr[\text{bad}] \leq 2^{-\lambda} \cdot \left(\underbrace{K}_{\text{Global setup phase}} + \underbrace{q_{\text{AS}} \cdot (L+2)}_{\text{Authority setup queries}} + \underbrace{q_{\text{KG}} \cdot (L_{\text{sk}}+1)}_{\text{Key generation queries}} + \underbrace{q_{\text{BGGM}}}_{\text{Generic group queries}} + \underbrace{M \cdot (L+3) + 2}_{\text{Challenge phase}} \right)^2.$$

Finally, recall that $L = \text{poly}_1(\lambda) \leq \lambda^{c_1}$ for all $\lambda \geq d_1$, $K = \text{poly}_2(\lambda) \leq \lambda^{c_2}$ for all $\lambda \geq d_2$,⁴ $q_{AS} = \text{poly}_3(\lambda) \leq \lambda^{c_3}$ for all $\lambda \geq d_3$, $q_{KG} = \text{poly}_4(\lambda) \leq \lambda^{c_4}$ for all $\lambda \geq d_4$, $q_{BGGM} = \text{poly}_5(\lambda) \leq \lambda^{c_5}$ for all $\lambda \geq d_5$, and $M = \text{poly}_6(\lambda) \leq \lambda^{c_6}$ for all $\lambda \geq d_6$ for some $c_1, c_2, c_3, c_4, c_5, c_6, d_1, d_2, d_3, d_4, d_5, d_6 \in \mathbb{N}$. Furthermore, $L_{sk} \leq L$. Let $c = \max(c_1, c_2, c_3, c_4, c_5, c_6)$ and $d = \max(d_1, d_2, d_3, d_4, d_5, d_6)$. We have that

$$\begin{aligned}
\Pr[\text{bad}] &\leq 2^{-\lambda} \cdot \left(\lambda^{c_2} + \lambda^{c_3} (\lambda^{c_1} + 2) + \lambda^{c_4} (\lambda^{c_1} + 1) + \lambda^{c_5} + \lambda^{c_6} (\lambda^{c_1} + 3) + 2 \right)^2 && \text{for } \lambda \geq d \\
&\leq 2^{-\lambda} \cdot \left(3\lambda^{2c} + 8\lambda^c + 2 \right)^2 && \text{for } \lambda \geq d \\
&\leq 2^{-\lambda} \cdot (16\lambda^{2c})^2 && \text{for } \lambda \geq d \\
&= 2^{8-\lambda+4c \log \lambda} && \text{for } \lambda \geq d \\
&\leq 2^{-\frac{1}{2}\lambda} && \text{for } \lambda \geq \max(d, 256c^2, 32).
\end{aligned}$$

It follows that there exists some negligible function $\text{negl}(\cdot)$ such that

$$|\Pr[\text{Hyb}_1^{(b)}(\lambda)] - \Pr[\text{Hyb}_2^{(b)}(\lambda)]| \leq \text{negl}(\lambda). \quad \square$$

Lemma 4.12. For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_2^{(b)}(\lambda)] = \Pr[\text{Hyb}_3^{(b)}(\lambda)]$.

Proof. The two hybrids are identical except that, in hybrid $\text{Hyb}_3^{(b)}$, the challenger additionally maintains adversary and challenger formal polynomial tables $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ respectively along with the set of corrupt authorities Z , and it answers a zero-test query on a handle $\langle h \rangle$ at time t by testing if $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ for the challenger's formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$. In hybrid $\text{Hyb}_2^{(b)}$, the challenger instead tested whether $x = 0$ for the exponent x where $\mathcal{T}(\langle h \rangle) = (x, v)$. Because neither table is given to the adversary $\mathcal{A}$ and neither is used to answer any other oracle query, we need only argue that these two ways of answering a zero-test query always agree. We do so by showing that the challenger's formal polynomial $\tilde{h}$ associated with a handle $\langle h \rangle$ always evaluates to the exponent x held in $\mathcal{T}(\langle h \rangle)$.

Before continuing with the analysis, we first recall the entries the challenger adds to $\tilde{\mathcal{T}}$ for the handles it produces over the course of the experiment:

$$\begin{aligned}
\mathbb{G}_1 \left\{ \begin{array}{ll} \langle y_\zeta \rangle \mapsto \tilde{y}_\zeta & \text{for } \zeta \in [q_{AS}] \\ \langle y_{i,\zeta} \rangle \mapsto \tilde{y}_{i,\zeta} & \text{for } (i, \zeta) \in [L] \times [q_{AS}] \\ \langle c_{3,x}^* \rangle \mapsto \tilde{r}_x & \text{for } x \in [\ell] \end{array} \right. & \quad \langle c_{4,x}^* \rangle \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{w}} + \tilde{r}_x \tilde{y}_{a_x}) \quad \text{for } x \in [\ell] \\
& \quad \langle c_{5,x,i}^* \rangle \mapsto \begin{cases} \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,a_x}) & i \in S_x^* \\ \tilde{r}'_{x,i} & \text{otherwise} \end{cases} \quad \text{for } (x, i) \in [\ell] \times [L], \\
\mathbb{G}_2 \left\{ \begin{array}{ll} \langle u_j \rangle \mapsto \tilde{u}_j & \text{for } j \in [K] \\ \langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi} & \text{for } \xi \in [q_{KG}], i \in S_\xi \end{array} \right. & \quad \langle \delta_\xi \rangle \mapsto \text{Corr}_Z \left(\tilde{\alpha}_\xi + \tilde{y}_\xi \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi} \right) \quad \text{for } \xi \in [q_{KG}], \\
\mathbb{G}_T \left\{ \begin{array}{ll} \langle \alpha_\zeta \rangle \mapsto \tilde{\alpha}_\zeta & \text{for } \zeta \in [q_{AS}] \\ \langle c_1^* \rangle \mapsto \tilde{s} + \tilde{m}_b^* & \end{array} \right. & \quad \langle c_{2,x}^* \rangle \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{v}} + \tilde{r}_x \tilde{\alpha}_{a_x}) \quad \text{for } x \in [\ell],
\end{aligned} \tag{4.1}$$

where msk_ζ is the authority secret key where $\zeta = \zeta(\xi)$ is the index of the authority setup query on the attribute a_ξ of the ξ^{th} key-generation query, $\tilde{m}_b^* = \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ is the challenger's formal polynomial associated with the challenge message handle $\langle m_b^* \rangle$, and, for each row $x \in [\ell]$ of the challenge policy, $a_x = \rho^*(x)$ and $\tilde{\alpha}_{a_x} = \tilde{\mathcal{T}}(\langle \alpha_{a_x} \rangle)$, $\tilde{y}_{a_x} = \tilde{\mathcal{T}}(\langle y_{a_x} \rangle)$, and $\tilde{y}_{i,a_x} = \tilde{\mathcal{T}}(\langle y_{i,a_x} \rangle)$ are the challenger's formal polynomials associated with the components of the authority public key mpk_{a_x} used to encrypt.

We also recall that each authority-corruption query replaces every entry $\langle h \rangle \mapsto \tilde{h}$ of $\tilde{\mathcal{T}}$ with $\langle h \rangle \mapsto \text{Corr}_Z(\tilde{h})$ for the updated set Z of corrupt authorities, where Corr_Z maps the variables $\tilde{\alpha}_\zeta$, $\tilde{y}_\zeta$, and $\tilde{y}_{i,\zeta}$ of each corrupt authority $\zeta \in Z$ to α_ζ , y_ζ , and $y_{i,\zeta}$ respectively. Furthermore, the assignment ϕ_t maps each challenger variable to its corresponding sampled value, provided the value has been sampled by time t (otherwise, it is mapped to 0). Note that, because

⁴ L and K are polynomial in λ and M , which is itself polynomial in λ .

ϕ_t also maps the variables $\tilde{\alpha}_\zeta$, $\tilde{y}_\zeta$, and $\tilde{y}_{i,\zeta}$ to α_ζ , y_ζ , and $y_{i,\zeta}$ respectively (i.e., the same values Corr_Z itself substitutes for the corrupt authorities), we accordingly have that $\text{Eval}_{\phi_t}(\text{Corr}_Z(\tilde{h})) = \text{Eval}_{\phi_t}(\tilde{h})$.

We now argue that $\text{Eval}_{\phi_t}(\tilde{h}) = x$ for every handle $\langle h \rangle$ with $\mathcal{T}(\langle h \rangle) = (x, v)$ and every time $t \geq t_h$, where t_h denotes the time at which $\langle h \rangle$ is created. For this, it suffices to show that $\text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$, as $\text{Eval}_{\phi_t}(\tilde{h}) = \text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$ for all $t \geq t_h$. This is because once an actual exponent is sampled and stored in $\mathcal{T}(\langle h \rangle)$, it is never reassigned; furthermore, the only subsequent modification made to $\tilde{\mathcal{T}}(\langle h \rangle)$ is the ring homomorphism Corr_Z on an authority-corruption query, which preserves evaluation as argued above.

Our analysis breaks down into a case covering handles mapping to values directly sampled by the challenger (i.e., handles output by the challenger during setup, authority setup, key-generation queries, and the challenge phase, excluding the challenge-ciphertext components that depend on handles supplied by $\mathcal{A}$: namely the handle $\langle c_1^* \rangle$ together with the handles $\langle c_{2,x}^* \rangle$, $\langle c_{4,x}^* \rangle$, and $\langle c_{5,x,i}^* \rangle$ for $i \in S_x^*$ for each row $x \in [\ell]$ with $a_x \notin A$, where A denotes the attributes queried to authority setup prior to the challenge phase) which we call *directly generated handles*, and a case covering handles mapping to values computed from values stored in other handles (i.e., handles produced by generic group oracle queries as well as the message-dependent and malicious-authority-dependent challenge-ciphertext components listed previously) which we call *derived handles*.

- **Case I: $\langle h \rangle$ is directly generated:** For this case, we can see from Eq. (4.1) that, by construction, each handle produced by the challenger during these phases maps directly to the polynomial that, when substituted with the appropriate variables (as is done when executing $\text{Eval}_{\phi_{t_h}}(\tilde{h})$), exactly computes what the challenger computes to produce these exponents. We accordingly only need to argue that the appropriate variables have been sampled before time t_h , as they are otherwise substituted with 0.

For handles $\langle u_j \rangle$, $\langle \alpha_\zeta \rangle$, $\langle y_\zeta \rangle$, $\langle y_{i,\zeta} \rangle$, $\langle \beta_{i,\xi} \rangle$, and $\langle c_{3,x}^* \rangle$, the challenger adds the sampled exponents u_j , α_ζ , y_ζ , $y_{i,\zeta}$, $\beta_{i,\xi}$, and r_x to $\mathcal{T}$ directly before adding the challenger formal polynomials $\tilde{u}_j$, $\tilde{\alpha}_\zeta$, $\tilde{y}_\zeta$, $\tilde{y}_{i,\zeta}$, $\tilde{\beta}_{i,\xi}$, and $\tilde{r}_x$ to $\tilde{\mathcal{T}}$.

Next, for a handle $\langle \delta_\xi \rangle$ which is sampled during a key-generation query, the challenger computes the exponent $\delta_\xi = \alpha_\zeta + y_\zeta h_\xi + \sum_{i \in S_\xi} y_{i,\zeta} \beta_{i,\xi}$ for $h_\xi = \sum_{j \in [K]} \text{gid}_\xi^{j-1} u_j$, and the values $u_1, \dots, u_K$, α_ζ , y_ζ , and $y_{i,\zeta}, \beta_{i,\xi}$ for $i \in S_\xi$ were all sampled prior in global setup, in the authority-setup query for a_ξ (because $\mathcal{A}$ is admissible), or earlier in the key-generation query in question.

For a handle $\langle c_{5,x,i}^* \rangle$ falling into this case (i.e., $\rho^*(x) = a_x$ where $a_x \in A$ was output by the ζ_x^{th} authority-setup query or where $i \notin S_x^*$), recall that the challenger computes the exponent $c_{5,x,i}^* = r_x y_{i,a_x}$ if $i \in S_x^*$ and it samples $c_{5,x,i}^* \xleftarrow{R} \mathbb{Z}_p$ otherwise, and that this occurs in the challenge phase. In the former case, this is mapped to the challenger formal polynomial $\text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,a_x})$, which is equal to $\tilde{r}_x \tilde{y}_{i,a_x}$ if the authority has not been corrupted and $y_{i,a_x} \tilde{r}_x$ if the authority has been corrupted. Observe that r_x was already sampled in the challenge phase, and y_{i,a_x} (if used) was sampled in the authority-setup query for a_x (in particular, the adversary must provide authority public-key handles for attributes on which an authority-setup query has not been made, putting those all in Case II). In the latter case, the entry is $\tilde{r}'_{x,i}$ which is added to $\mathcal{T}$ directly after sampling $c_{5,x,i}^*$.

Finally, for the handles $\langle c_{2,x}^* \rangle$ and $\langle c_{4,x}^* \rangle$ falling into this case, the challenger adds entries mapping the handles to $\text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{v}} + \tilde{r}_x \tilde{\alpha}_{\zeta_x})$ and $\text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{w}} + \tilde{r}_x \tilde{y}_{\zeta_x})$ respectively, where $\tilde{\mathbf{v}} = (\tilde{s}, \tilde{v}_2, \dots, \tilde{v}_n)$ and $\tilde{\mathbf{w}} = (0, \tilde{w}_2, \dots, \tilde{w}_n)$. Observe that the corresponding values $s, v_2, \dots, v_n, w_2, \dots, w_n$, and r_x were all sampled prior in the challenge phase, and the values α_{ζ_x} and y_{ζ_x} were sampled in the authority-setup query for a_x (following the same argument as above).

- **Case II: $\langle h \rangle$ is derived:** We use induction on the order in which the handles were sampled.

The base case is the first handle created over the course of the experiment, which is a handle $\langle u_j \rangle$ that the challenger produces during setup. This is directly covered by Case I.

For the inductive step, we first consider each of the three generic group oracles separately. For the BGGM.Add and BGGM.Pair oracles, we write $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$, and we note that the handles $\langle h_1 \rangle$ and $\langle h_2 \rangle$ must be sampled before $\langle h \rangle$ for the oracle to not have aborted. Thus, by the induction hypothesis, we have that $\text{Eval}_{\phi_{t_h}}(h_1) = x_1$ and $\text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_2$ at the time t_h .

- $\text{BGGM.Encode}(x, v)$: The challenger adds the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, thus $\text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$.
- $\text{BGGM.Add}(\langle h_1 \rangle, \langle h_2 \rangle)$: The challenger adds the entry $\langle h \rangle \mapsto (x_1 + x_2, v_1)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto \tilde{h}_1 + \tilde{h}_2$ to $\tilde{\mathcal{T}}$. Notably, every variable in $\tilde{h}_1 + \tilde{h}_2$ appears in $\tilde{h}_1$ or in $\tilde{h}_2$, and has thus already been sampled by time t_{h_1} or t_{h_2} where $t_{h_1}, t_{h_2} < t_h$. Furthermore, because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{h}_1 + \tilde{h}_2) = \text{Eval}_{\phi_{t_h}}(\tilde{h}_1) + \text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_1 + x_2.$$

- $\text{BGGM.Pair}(\langle h_1 \rangle, \langle h_2 \rangle)$: The challenger adds the entry $\langle h \rangle \mapsto (x_1 x_2, T)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto \tilde{h}_1 \cdot \tilde{h}_2$ to $\tilde{\mathcal{T}}$. Notably, every variable in $\tilde{h}_1 \cdot \tilde{h}_2$ appears in $\tilde{h}_1$ or in $\tilde{h}_2$, and has thus already been sampled by time t_{h_1} or t_{h_2} where $t_{h_1}, t_{h_2} < t_h$. Furthermore, because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{h}_1 \cdot \tilde{h}_2) = \text{Eval}_{\phi_{t_h}}(\tilde{h}_1) \cdot \text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_1 x_2.$$

Next, we analyze the handle $\langle c_1^* \rangle$, which the challenger produces during the challenge phase by executing $\text{BGGM.Encode}(s + m_b^*, T)$ to add the entries $\langle c_1^* \rangle \mapsto (s + m_b^*, T)$ to $\mathcal{T}$ and $\langle c_1^* \rangle \mapsto \tilde{s} + \tilde{m}_b^*$ to $\tilde{\mathcal{T}}$, where $\mathcal{T}(\langle m_b^* \rangle) = (m_b^*, T)$ is the entry for the challenge message handle. As above, the value s was sampled prior in the challenge phase. Furthermore, the challenge message handle $\langle m_b^* \rangle$ must have been sampled before the adversary $\mathcal{A}$ submits it and thus before $\langle c_1^* \rangle$ is sampled. By the induction hypothesis $\text{Eval}_{\phi_{t_h}}(\tilde{m}_b^*) = m_b^*$, and because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{s} + \tilde{m}_b^*) = s + m_b^*.$$

Finally, we analyze the handles $\langle c_{2,x}^* \rangle$, $\langle c_{4,x}^* \rangle$, and $\langle c_{5,x,i}^* \rangle$ for $i \in S_x^*$ for each row $x \in [\ell]$ with $a_x \notin A$, which the challenger produces during the challenge phase by encoding the exponents $\mathbf{m}_x^{*\top} \mathbf{v} + r_x \alpha_{a_x}$, $\mathbf{m}_x^{*\top} \mathbf{w} + r_x y_{a_x}$, and $r_x \gamma_{i,a_x}$, where it retrieves the exponents α_{a_x} , y_{a_x} , and γ_{i,a_x} from the entries in $\mathcal{T}$ for the authority public-key handles $\langle \alpha_{a_x} \rangle$, $\langle y_{a_x} \rangle$, and $\langle \gamma_{i,a_x} \rangle$ supplied by $\mathcal{A}$ in this phase. As above, the values $s, v_2, \dots, v_n, w_2, \dots, w_n$, and r_x were all sampled prior in the challenge phase. Furthermore, each supplied handle must have been sampled before $\mathcal{A}$ submits it and thus before the corresponding challenge-ciphertext component is sampled. Thus, by the induction hypothesis, $\text{Eval}_{\phi_{t_h}}(\tilde{\alpha}_{a_x}) = \alpha_{a_x}$, $\text{Eval}_{\phi_{t_h}}(\tilde{y}_{a_x}) = y_{a_x}$, and $\text{Eval}_{\phi_{t_h}}(\tilde{\gamma}_{i,a_x}) = \gamma_{i,a_x}$, and because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\begin{aligned} \text{Eval}_{\phi_{t_h}}(\text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{v}} + \tilde{r}_x \tilde{\alpha}_{a_x})) &= \text{Eval}_{\phi_{t_h}}(\mathbf{m}_x^{*\top} \tilde{\mathbf{v}} + \tilde{r}_x \tilde{\alpha}_{a_x}) = \mathbf{m}_x^{*\top} \mathbf{v} + r_x \alpha_{a_x}, \\ \text{Eval}_{\phi_{t_h}}(\text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{w}} + \tilde{r}_x \tilde{y}_{a_x})) &= \text{Eval}_{\phi_{t_h}}(\mathbf{m}_x^{*\top} \tilde{\mathbf{w}} + \tilde{r}_x \tilde{y}_{a_x}) = \mathbf{m}_x^{*\top} \mathbf{w} + r_x y_{a_x}, \\ \text{Eval}_{\phi_{t_h}}(\text{Corr}_Z(\tilde{r}_x \tilde{\gamma}_{i,a_x})) &= \text{Eval}_{\phi_{t_h}}(\tilde{r}_x \tilde{\gamma}_{i,a_x}) = r_x \gamma_{i,a_x}. \end{aligned}$$

As we have shown above, the challenger's formal polynomial $\tilde{h}$ associated with a handle $\langle h \rangle$ always evaluates to the exponent x held in $\mathcal{T}(\langle h \rangle) = (x, v)$, and the lemma follows. Before continuing, we remark that we can also consider a mapping from the adversary's formal polynomials to challenger formal polynomials given by:

$$\begin{aligned} \mathbb{G}_1 \begin{cases} \hat{y}_\zeta \mapsto \text{Corr}_Z(\tilde{y}_\zeta) & \text{for } \zeta \in [q_{AS}] \\ \hat{y}_{i,\zeta} \mapsto \text{Corr}_Z(\tilde{y}_{i,\zeta}) & \text{for } (i, \zeta) \in [L] \times [q_{AS}] \\ \hat{c}_{3,x}^* \mapsto \tilde{r}_x & \text{for } x \in [\ell] \end{cases} & \quad \hat{c}_{4,x}^* \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{w}} + \tilde{r}_x \tilde{y}_{a_x}) \quad \text{for } x \in [\ell] \\ & \quad \hat{c}_{5,x,i}^* \mapsto \begin{cases} \text{Corr}_Z(\tilde{r}_x \tilde{\gamma}_{i,a_x}) & i \in S_x^* \\ \tilde{r}'_{x,i} & \text{otherwise} \end{cases} \quad \text{for } x \in [\ell], i \in [L], \\ \mathbb{G}_2 \begin{cases} \hat{u}_j \mapsto \tilde{u}_j & \text{for } j \in [K] \\ \hat{\beta}_{i,\xi} \mapsto \tilde{\beta}_{i,\xi} & \text{for } \xi \in [q_{KG}], i \in S_\xi \end{cases} & \quad \hat{\delta}_\xi \mapsto \text{Corr}_Z\left(\tilde{\alpha}_\xi + \tilde{y}_\xi \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi}\right) \quad \text{for } \xi \in [q_{KG}], \\ \mathbb{G}_T \begin{cases} \hat{\alpha}_\zeta \mapsto \text{Corr}_Z(\tilde{\alpha}_\zeta) & \text{for } \zeta \in [q_{AS}] \\ \hat{c}_1^* \mapsto \tilde{s} + \tilde{m}_b^* \end{cases} & \quad \hat{c}_{2,x}^* \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\top} \tilde{\mathbf{v}} + \tilde{r}_x \tilde{\alpha}_{a_x}) \quad \text{for } x \in [\ell], \end{aligned} \tag{4.2}$$

where Z is the set of corrupt authorities at the time considered and where $\tilde{m}_b^*$, $\tilde{\alpha}_{a_x}$, $\tilde{y}_{a_x}$, and $\tilde{y}_{i,a_x}$ are as in Eq. (4.1). Note that the variables indexed by rows $x \in [M] \setminus [\ell]$ never appear in $\tilde{\mathcal{T}}$, so we do not need to consider them (i.e., they can all map to 0).

We can extend this mapping to a ring homomorphism $\text{MapP}: \mathbb{Z}_p[\hat{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ in the natural way, such that $\tilde{h} = \text{MapP}(\hat{h})$ for every handle $\langle h \rangle$, where $\hat{h} = \hat{\mathcal{T}}(\langle h \rangle)$ and $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ are the adversary's and the challenger's formal polynomials associated with $\langle h \rangle$. This holds by construction for the handles the challenger produces, as can be seen by inspection of Eq. (4.2), and it is preserved by the generic group oracles, as each of them updates $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ with the same ring operation (a constant polynomial for encode oracle queries, and the sum or the product of the entries of the queried handles for add and pair oracle queries), which MapP respects. It is furthermore preserved by the authority-corruption queries, as these never modify $\hat{\mathcal{T}}$, while Corr_Z is applied to every entry of $\tilde{\mathcal{T}}$, which amounts to replacing MapP with $\text{Corr}_Z \circ \text{MapP}$.

Note that because MapP is a ring homomorphism, it acts on the adversary's formal polynomial term by term (i.e., each term of $\hat{h}$ is mapped to the product of the challenger polynomials of the adversary variables it comprises, and $\tilde{h}$ is the sum of the resulting polynomials). We accordingly say that a term of $\hat{h}$ *induces* the monomials over $\tilde{\mathcal{V}}$ that appear in its image, and, when analyzing the subsequent hybrids, we will study $\tilde{h}$ by asking which terms of $\hat{h}$ can induce a given monomial. Notably, several terms of $\hat{h}$ may induce the same monomial, in which case their contributions to $\tilde{h}$ may cancel. $\square$

Lemma 4.13. *There exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,*

$$|\Pr[\text{Hyb}_3^{(b)}(\lambda)] - \Pr[\text{Hyb}_4^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The hybrids are equivalent except for the event where the zero-test oracle is queried on a handle $\langle h \rangle$ associated with challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ where $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ (so $\text{Hyb}_3^{(b)}$ returns “zero”) but $\tilde{h} \neq 0$ (so $\text{Hyb}_4^{(b)}$ returns “non-zero”). Let bad denote this event. For a particular zero-test query on handle $\langle h' \rangle$ at time t' , we first observe that the associated polynomial $\tilde{h}'$ is at most degree 4. This is because the construction gives out elements as follows:

- In global setup, authority setup, and key generation, the challenger's formal polynomials associated with handles in the left group $\mathbb{G}_1$ have degree at most 1 and those associated with handles in the right group $\mathbb{G}_2$ have degree at most 2, so any handle in the target group $\mathbb{G}_T$ produced before the challenge phase (including $\langle \alpha_\zeta \rangle$, the challenge message handle $\langle m_b^* \rangle$, and the authority public-key handles $\langle \alpha_{a^*} \rangle$ supplied by $\mathcal{A}$) has degree at most 3.
- In encryption, the handles $\langle c_{3,x}^* \rangle$, $\langle c_{4,x}^* \rangle$, and $\langle c_{5,x,i}^* \rangle$ in the left group $\mathbb{G}_1$ have degree at most 2 (as $\langle y_{a_x} \rangle$ and $\langle y_{i,a_x} \rangle$ have degree at most 1, even if supplied by $\mathcal{A}$), and the handles $\langle c_1^* \rangle$ and $\langle c_{2,x}^* \rangle$ in $\mathbb{G}_T$ have degree at most 4 (as $\langle m_b^* \rangle$ and $\langle \alpha_{a_x} \rangle$ have degree at most 3 by the case above).

Consequently, every handle in $\mathbb{G}_1$ or $\mathbb{G}_2$ has degree at most 2, so $\mathcal{A}$ cannot pair to produce any polynomial of degree greater than 4, and every component in $\mathbb{G}_T$ has degree at most 4. Furthermore, applying Corr_Z on an authority-corruption query never increases the degree.

As hybrids $\text{Hyb}_3^{(b)}$ and $\text{Hyb}_4^{(b)}$ are identical until bad , it suffices to bound $\Pr[\text{bad}]$ in either hybrid [BR06]. In $\text{Hyb}_4^{(b)}$, the zero-test oracle answers by checking whether $\tilde{h} \equiv 0$, so the only exponents from $\mathcal{T}$ given to $\mathcal{A}$ are the authority secret keys $\text{msk}_\zeta = (\alpha_\zeta, y_\zeta, \{y_{i,\zeta}\}_{i \in [L]})$ of corrupt authorities ζ . Consider a zero-test query on a handle $\langle h' \rangle$ at time t' , and let Z be the set of corrupt authorities at time t' . Then $\mathcal{A}$'s view up to time t' , and hence $\tilde{h}'$, is independent of the values of all challenger variables other than $\tilde{\alpha}_\zeta$, $\tilde{y}_\zeta$, and $\tilde{y}_{i,\zeta}$ for $\zeta \in Z$, and none of these occur in $\tilde{h}'$, as Corr_Z has already replaced them with constants. Moreover, each variable occurring in $\tilde{h}'$ is added to $\tilde{\mathcal{T}}$ only when the challenger samples the corresponding exponent uniformly from $\mathbb{Z}_p$, so $\phi_{t'}$ assigns it this sampled value (rather than 0). We may therefore apply the Schwartz-Zippel lemma (Lemma 2.3) to each zero-test query as though $\tilde{h}'$ were fixed in advance. The lemma follows by applying the union bound and using the fact that $q_{ZT} = \text{poly}_3(\lambda)$ and $p \geq 2^\lambda$.

$$\Pr[\text{bad}] \leq q_{ZT} \cdot \Pr[\text{Eval}_{\phi_{t'}}(\tilde{h}') = 0 \wedge \tilde{h}' \neq 0] \leq \frac{4 \cdot q_{ZT}}{p} = \text{negl}(\lambda). \quad \square$$

Recall that the behavior of the remaining hybrids is dependent on which authorities the adversary $\mathcal{A}$ has corrupted by a given zero-test query. In particular, let Z_t denote the set of authority-setup indices for corrupted authorities Z at the time t and Z_χ denote this set at the time of the χ^{th} post-challenge zero-test query. We define $X_{\chi,\text{hon}}$ to be the set of rows $x \in [\ell]$ where $\rho^*(x) = a_\zeta$ for some $\zeta \notin Z_\chi$ (i.e., x is not for a corrupted attribute) and $X_{\chi,\text{cor}} = [\ell] \setminus X_{\chi,\text{hon}}$. Furthermore, we let $X_{0,\text{hon}}$ and $X_{0,\text{cor}}$ denote these sets at the start of the post-challenge query phase.

Recall that we also defined the following procedure given a set $X \subseteq [\ell]$. For each $x \in X$, the challenger samples a set $S_x^* \xleftarrow{R} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. Let $\text{SubS}_{X,Z} : \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ be the ring homomorphism induced by

$$\tilde{r}'_{x,i} \mapsto \text{Corr}_Z(\tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle y_{i,a_x} \rangle)) \quad \text{for } x \in X, i \in S_x^* \quad \text{and} \quad \tilde{v} \mapsto \tilde{v} \quad \text{otherwise.}$$

We say that the challenger *samples and substitutes* (X, Z) if it replaces every entry $\langle h \rangle \mapsto \tilde{h}'$ of $\tilde{\mathcal{T}}$ with the entry $\langle h \rangle \mapsto \text{SubS}_{X,Z}(\tilde{h}')$ (i.e., the $\tilde{r}'_{x,i}$ components are substituted with $\text{Corr}_Z(\tilde{r}_x \cdot \tilde{\mathcal{T}}(\langle y_{i,a_x} \rangle))$ for $x \in X$ and $i \in S_x^*$). To prove the remaining lemmas, we will require the following claim on the properties of the SubS ring homomorphism.

Claim 4.14. *The ring homomorphism SubS satisfies the following properties:*

- **Time independence:** For every $X \subseteq X_{\chi,\text{hon}}$, at every time t at or after the χ^{th} post-challenge zero-test query (or the start of the post-challenge query phase if $\chi = 0$), and every $Z \supseteq Z_t$, we have that

$$\text{SubS}_{X,Z} : \tilde{r}'_{x,i} \mapsto \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x}) \quad \text{for } x \in X, i \in S_x^* \quad \text{and} \quad \tilde{v} \mapsto \tilde{v} \quad \text{otherwise,}$$

i.e., although $\text{SubS}_{X,Z}$ is defined in terms of the entries $\tilde{\mathcal{T}}(\langle y_{i,a_x} \rangle)$, which may be rewritten by authority-corruption queries, it is the same ring homomorphism no matter when it is applied.

- **Corruption commutativity:** For every $X \subseteq X_{\chi,\text{hon}}$, at every time t at or after the χ^{th} post-challenge zero-test query (or the start of the post-challenge query phase if $\chi = 0$), and all $Z_t \subseteq Z \subseteq Z'$, we have that

$$\text{Corr}_{Z'} \circ \text{SubS}_{X,Z} = \text{SubS}_{X,Z'} \circ \text{Corr}_Z,$$

i.e., corrupting Z' after performing the SubS substitution with the smaller corruption set Z is the same as performing the SubS substitution with the updated corruption set Z' after corrupting Z' .

- **Parallel composition:** For all disjoint $X_1, X_2 \subseteq X_{\chi,\text{hon}}$, at every time t at or after the χ^{th} post-challenge zero-test query (or the start of the post-challenge query phase if $\chi = 0$), and every $Z \supseteq Z_t$, we have that

$$\text{SubS}_{X_1 \cup X_2, Z} = \text{SubS}_{X_2, Z} \circ \text{SubS}_{X_1, Z}$$

i.e., performing the SubS substitution for $X_1 \cup X_2$ at once is the same as first performing the SubS substitution for X_1 and then performing the SubS substitution for X_2 .

- **Invariance:** For every $X \subseteq [\ell]$ and Z , if $\tilde{h} \in \mathbb{Z}_p[\tilde{\mathcal{V}}]$ does not have any of the variables $\{\tilde{r}'_{x,i}\}_{(x,i) \in X \times [L]}$, then $\text{SubS}_{X,Z}(\tilde{h}) = \tilde{h}$, i.e., substituting for X only affects the terms involving the variables $\tilde{r}'_{x,i}$ for rows $x \in X$.

Proof. We prove each property one-by-one. Throughout, let t be a time on or after the χ^{th} post-challenge zero-test query (or the start of the post-challenge query phase if $\chi = 0$), and note that $Z_\chi \subseteq Z_t$ (as elements are only ever added to the corruption set Z).

- **Time independence:** For every row $x \in X_{\chi,\text{hon}}$, we have that a_x was previously queried to authority setup and $\zeta_x \notin Z_\chi$, so $\tilde{\mathcal{T}}(\langle y_{i,a_x} \rangle) = \text{Corr}_{Z_t}(\tilde{y}_{i,\zeta_x})$ at time t . It thus follows that, for every $X \subseteq X_{\chi,\text{hon}}$ and every $Z \supseteq Z_t$,

$$\text{SubS}_{X,Z} : \tilde{r}'_{x,i} \mapsto \text{Corr}_Z(\tilde{r}_x \cdot \text{Corr}_{Z_t}(\tilde{y}_{i,\zeta_x})) = \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x}) \quad \text{for } x \in X, i \in S_x^* \quad \text{and} \quad \tilde{v} \mapsto \tilde{v} \quad \text{otherwise,}$$

following from the fact that $\text{Corr}_Z \circ \text{Corr}_{Z_t} = \text{Corr}_Z$ whenever $Z_t \subseteq Z$.

- **Corruption commutativity:** Let $X \subseteq X_{\mathcal{X},\text{hon}}$ and $Z_t \subseteq Z \subseteq Z'$. As both sides are ring homomorphisms, it suffices to check that they agree on each variable $\tilde{v} \in \tilde{\mathcal{V}}$. First observe that if $\tilde{v} = \tilde{r}'_{x,i}$ with $x \in X$ and $i \in S_x^*$, we have

$$\text{Corr}_{Z'}(\text{Sub}_{X,Z}(\tilde{r}'_{x,i})) = \text{Corr}_{Z'}(\text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x})) = \text{Corr}_{Z'}(\tilde{r}_x \tilde{y}_{i,\zeta_x}) = \text{Sub}_{X,Z'}(\tilde{r}'_{x,i}) = \text{Sub}_{X,Z'}(\text{Corr}_{Z'}(\tilde{r}'_{x,i})),$$

following from time independence, the fact that $\text{Corr}_{Z'} \circ \text{Corr}_Z = \text{Corr}_{Z'}$ whenever $Z \subseteq Z'$, and the fact that $\text{Corr}_{Z'}(\tilde{r}'_{x,i}) = \tilde{r}'_{x,i}$ (as $\text{Corr}_{Z'}$ only substitutes the variables of corrupt authorities). For every other variable $\tilde{v}$, we have $\text{Sub}_{X,Z}(\tilde{v}) = \tilde{v}$ by definition, while $\text{Corr}_{Z'}(\tilde{v})$ is either $\tilde{v}$ itself or a constant, and is hence mapped to itself by $\text{Sub}_{X,Z'}$. It follows that,

$$\text{Corr}_{Z'}(\text{Sub}_{X,Z}(\tilde{v})) = \text{Corr}_{Z'}(\tilde{v}) = \text{Sub}_{X,Z'}(\text{Corr}_{Z'}(\tilde{v})).$$

- **Parallel composition:** Let $X_1, X_2 \subseteq X_{\mathcal{X},\text{hon}}$ be disjoint and $Z \supseteq Z_t$. As both sides are ring homomorphisms, it suffices to check that they agree on each variable $\tilde{v} \in \tilde{\mathcal{V}}$. First observe that if $\tilde{v} = \tilde{r}'_{x,i}$ with $x \in X_1$ and $i \in S_x^*$, we have

$$\text{Sub}_{X_2,Z}(\text{Sub}_{X_1,Z}(\tilde{r}'_{x,i})) = \text{Sub}_{X_2,Z}(\text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x})) = \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x}) = \text{Sub}_{X_1 \cup X_2,Z}(\tilde{r}'_{x,i}),$$

following from time independence and the fact that $\text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x})$ does not have any variable $\tilde{r}'_{x',i'}$. Similarly, if $\tilde{v} = \tilde{r}'_{x,i}$ with $x \in X_2$ and $i \in S_x^*$, we have

$$\text{Sub}_{X_2,Z}(\text{Sub}_{X_1,Z}(\tilde{r}'_{x,i})) = \text{Sub}_{X_2,Z}(\tilde{r}'_{x,i}) = \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x}) = \text{Sub}_{X_1 \cup X_2,Z}(\tilde{r}'_{x,i}),$$

following from time independence and the fact that $x \notin X_1$ (as X_1 and X_2 are disjoint). For every other variable $\tilde{v}$, we have $\text{Sub}_{X_1,Z}(\tilde{v}) = \text{Sub}_{X_2,Z}(\tilde{v}) = \text{Sub}_{X_1 \cup X_2,Z}(\tilde{v}) = \tilde{v}$ by definition. It follows that,

$$\text{Sub}_{X_2,Z}(\text{Sub}_{X_1,Z}(\tilde{v})) = \tilde{v} = \text{Sub}_{X_1 \cup X_2,Z}(\tilde{v}).$$

- **Invariance:** By definition, $\text{Sub}_{X,Z}$ maps every variable other than $\{\tilde{r}'_{x,i}\}_{(x,i) \in X \times [L]}$ to itself. As $\text{Sub}_{X,Z}$ is a ring homomorphism and $\tilde{h}$ does not have any of these variables, it follows that $\text{Sub}_{X,Z}(\tilde{h}) = \tilde{h}$. $\square$

Lemma 4.15. For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_4^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,0,\text{int}}^{(b)}(\lambda)]$.

Proof. All that differs between the two hybrids is when the challenger samples the sets S_x^* for $x \in X_{0,\text{hon}}$ and performs the substitution (via the ring homomorphism $\text{Sub}_{X_{0,\text{hon}},Z_0}$, where Z_0 denotes the set of corrupt authorities at the start of the post-challenge query phase) on the entries of $\tilde{\mathcal{T}}$. Specifically, this step is performed at the start of the post-challenge query phase in hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$, while in hybrid $\text{Hyb}_4^{(b)}$ it is instead performed at the end of the challenge phase. That is, the challenger directly records the substituted entry $\langle c_{5,x,i}^* \rangle \mapsto \text{Corr}_{Z_0}(\tilde{r}_x \tilde{y}_{i,\zeta_x})$ for each $x \in X_{0,\text{hon}}$ and $i \in S_x^*$ in hybrid $\text{Hyb}_4^{(b)}$ instead of the entry $\langle c_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$ as is done in hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$ (for the rows $x \in X_{0,\text{cor}}$, both hybrids sample S_x^* and record the same entries during the challenge phase). We will argue that $\mathcal{A}$'s view is identical in both hybrids.

We first note that, starting in hybrid $\text{Hyb}_4^{(b)}$ (and for all remaining hybrids), the zero-test oracle answers a query on a handle $\langle h \rangle$ by testing whether $\tilde{h} \equiv 0$ (or alternatively not checking $\tilde{h}$ at all), and nothing derived from exponents from the table $\mathcal{T}$ is given to $\mathcal{A}$ other than the authority secret keys $\text{msk}_\zeta = (\alpha_\zeta, y_\zeta, \{y_{i,\zeta}\}_{i \in [L]})$ of *corrupt* authorities ζ , which are sampled independently of the sets S_x^* for $x \in X_{0,\text{hon}}$. It follows that the sets S_x^* for $x \in X_{0,\text{hon}}$ influence $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$.

Now, observe that $\mathcal{A}$'s view through the end of the challenge phase is identical in both hybrids, as the two hybrids differ only in the entries the challenger records in $\tilde{\mathcal{T}}$ for the handles $\langle c_{5,x,i}^* \rangle$. This is because $\mathcal{A}$ cannot make any generic group oracle queries between the end of the challenge phase and the substitution point at the start of the post-challenge query phase (notably, no handles are sampled and no zero-test queries are made). In particular, the handles present in $\tilde{\mathcal{T}}$ at the substitution point are exactly those present at the end of the challenge phase.

It remains to argue that the entries of $\tilde{\mathcal{T}}$ coincide in both hybrids from the substitution point onwards. Consider any handle $\langle h \rangle$ present in $\tilde{\mathcal{T}}$ and let $\tilde{h}'$ denote the entry it would have *at the same point in time* in the absence of any

substitution (i.e., the entry obtained by recording $\langle c_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$ for every $x \in X_{0,\text{hon}}$ and $i \in [L]$). Note that we have the requirement on time because $\tilde{h}'$ can be rewritten by the authority-corruption queries. We write $\text{SubS}_Z = \text{SubS}_{X_{0,\text{hon}},Z}$ (because this hybrid only samples and substitutes with $X_{0,\text{hon}}$) and let t_0 denote the substitution point. By time independence (Claim 4.14), at every time $t \geq t_0$ and for every $Z \supseteq Z_t$, SubS_Z maps $\tilde{r}'_{x,i} \mapsto \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x})$ for $x \in X_{0,\text{hon}}$ and $i \in S_x^*$, and every other variable to itself. We now argue, by induction on the steps that create or modify entries of $\tilde{\mathcal{T}}$ from the substitution point onwards, that $\tilde{\mathcal{T}}(\langle h \rangle) = \text{SubS}_{Z_t}(\tilde{h}')$ holds at every such time $t \geq t_0$, for every handle $\langle h \rangle$ present in $\tilde{\mathcal{T}}$, in hybrid $\text{Hyb}_4^{(b)}$ and in hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$, and thus the behavior of the hybrids is equivalent.

For the base case, consider a handle $\langle h \rangle$ that is stored in $\tilde{\mathcal{T}}$ at the end of the challenge phase (i.e., at the substitution point, where $Z_t = Z_0$). We begin by analyzing hybrid $\text{Hyb}_4^{(b)}$. If $\langle h \rangle = \langle c_{5,x,i}^* \rangle$ for some $x \in X_{0,\text{hon}}$ and $i \in [L]$, then hybrid $\text{Hyb}_4^{(b)}$ records the entry $\langle h \rangle \mapsto \text{Corr}_{Z_0}(\tilde{r}_x \tilde{y}_{i,\zeta_x})$ if $i \in S_x^*$ and $\langle h \rangle \mapsto \tilde{r}'_{x,i}$ otherwise, which in either case is exactly $\text{SubS}_{Z_0}(\tilde{r}'_{x,i}) = \text{SubS}_{Z_0}(\tilde{h}')$. Thus the entry $\langle h \rangle \mapsto \text{SubS}_{Z_0}(\tilde{h}')$ is added to $\tilde{\mathcal{T}}$. Otherwise, $\tilde{h}'$ does not contain any term involving any of the variables $\{\tilde{r}'_{x,i}\}_{(x,i) \in X_{0,\text{hon}} \times [L]}$. In particular, by inspection of Eq. (4.1), the entries $\langle c_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$ are the only ones the challenger adds that induce such a variable and no other entry the challenger adds during the challenge phase involves them, while every adversary-added entry is added by a generic group oracle query and is either a constant polynomial or a sum or product of entries created earlier. Hence, by invariance (Claim 4.14), $\text{SubS}_{Z_0}(\tilde{h}') = \tilde{h}'$, and so the entry recorded is again $\langle h \rangle \mapsto \text{SubS}_{Z_0}(\tilde{h}')$. Moving on to hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$, the base case holds by definition, as the substitution step replaces $\tilde{h}'$ in $\tilde{\mathcal{T}}$ with $\text{SubS}_{Z_0}(\tilde{h}')$.

For the inductive step, consider a creation or modification of $\tilde{\mathcal{T}}$ performed after the substitution point, and let t^- and t^+ denote the times immediately before and after it. Now suppose that $\tilde{\mathcal{T}}(\langle h \rangle) = \text{SubS}_{Z_{t^-}}(\tilde{h}')$ for all handles $\langle h \rangle$ in the table at time t^- . We go over each possible change to $\tilde{\mathcal{T}}$ below:

- **Post-challenge authority-setup queries:** On the ζ^{th} authority-setup query, the challenger adds the entries $\langle \alpha_\zeta \rangle \mapsto \tilde{\alpha}_\zeta$, $\langle y_\zeta \rangle \mapsto \tilde{y}_\zeta$, and $\langle y_{i,\zeta} \rangle \mapsto \tilde{y}_{i,\zeta}$ for each $i \in [L]$ to $\tilde{\mathcal{T}}$. The corrupted authority set is unchanged, so $Z_{t^+} = Z_{t^-}$, and no entry contains any of the variables $\{\tilde{r}'_{x,i}\}_{(x,i) \in X_{0,\text{hon}} \times [L]}$, and so, by invariance (Claim 4.14), all added entries are unaffected by the $\text{SubS}_{Z_{t^+}}$ ring homomorphism.
- **Post-challenge authority-corruption queries:** A corruption query replaces every entry $\langle h \rangle \mapsto \tilde{h}$ of $\tilde{\mathcal{T}}$ with $\langle h \rangle \mapsto \text{Corr}_{Z_{t^+}}(\tilde{h})$ for the updated set Z_{t^+} of corrupted authorities. Let $\tilde{h}'$ denote the entry $\langle h \rangle$ would have at time t^- in the absence of any substitution, so that the corresponding entry at time t^+ is $\text{Corr}_{Z_{t^+}}(\tilde{h}')$. As $Z_{t^-} \subseteq Z_{t^+}$, the induction hypothesis and corruption commutativity (Claim 4.14) give us, in both hybrids,

$$\tilde{\mathcal{T}}(\langle h \rangle) = \text{Corr}_{Z_{t^+}}(\text{SubS}_{Z_{t^-}}(\tilde{h}')) = \text{SubS}_{Z_{t^+}}(\text{Corr}_{Z_{t^+}}(\tilde{h}')).$$

- **Post-challenge key-generation queries:** On the ξ^{th} key-generation query, the challenger adds the entries $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ for each $i \in S_\xi$ and $\langle \delta_\xi \rangle \mapsto \text{Corr}_{Z_{t^+}}(\tilde{\alpha}_\xi + \tilde{y}_\xi \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\zeta} \tilde{\beta}_{i,\xi})$ to $\tilde{\mathcal{T}}$. Again $Z_{t^+} = Z_{t^-}$ and neither entry contains any of the variables $\{\tilde{r}'_{x,i}\}_{(x,i) \in X_{0,\text{hon}} \times [L]}$, so, by invariance (Claim 4.14), both are unaffected by the $\text{SubS}_{Z_{t^+}}$ ring homomorphism.
- **Post-challenge generic group oracle queries:** All three of these queries also leave the corrupt set unchanged, so $Z_{t^+} = Z_{t^-}$. The analysis of the generic group oracle queries is as follows:

- **BGGM.Encode(x, v):** The challenger adds the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, which is again unaffected by the $\text{SubS}_{Z_{t^+}}$ ring homomorphism.
- **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** Recall that the challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that $\text{SubS}_{Z_{t^+}}$ is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1) + \text{SubS}_{Z_{t^+}}(\tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1 + \tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}').$$

- **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** The challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that $\text{SubS}_{Z_{t^+}}$ is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1) \cdot \text{SubS}_{Z_{t^+}}(\tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1 \cdot \tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}').$$

In each case, we have that $\tilde{\mathcal{T}}(\langle h \rangle) = \text{SubS}_{Z_{t^+}}(\tilde{h}')$ at time t^+ in both hybrids, and thus $\mathcal{A}$'s view is identical in the two hybrids. It follows that

$$\Pr[\text{Hyb}_4^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,0,\text{int}}^{(b)}(\lambda)]. \quad \square$$

Lemma 4.16. *For $L - L_{\text{ct}} = \omega(\log \lambda)$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ and all $\chi \in [q_{\text{ZT}}]$,*

$$|\Pr[\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}(\lambda)] - \Pr[\text{Hyb}_{5,\chi}^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. All that differs between the two hybrids is the behavior of the zero-test oracle on the χ^{th} post-challenge query the adversary $\mathcal{A}$ makes. Specifically, for said zero-test query on handle $\langle h \rangle$, the oracle checks if the challenger's formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ is identically zero in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$, while it instead automatically returns “non-zero” when certain message- and challenge-set-dependent terms are present in the expanded adversary formal polynomial $\hat{h}_Z = \text{Expd}_{Z_\chi}(\hat{\mathcal{T}}(\langle h \rangle))$ (as defined in the hybrid) in hybrid $\text{Hyb}_{5,\chi}^{(b)}$. We will argue that, except with negligible probability, the challenger's formal polynomial $\tilde{h}$ in this case would anyway not be identically zero and the oracle would output “non-zero” in any case. We let $\tilde{h}$ denote the event that $\tilde{h}$ (corresponding to an $\hat{h}_Z$ that has the challenge message- or challenge-set-dependent terms) is identically zero.

Before continuing, we recall how the challenger's formal polynomials relate to the adversary's in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$. Let MapP_{int} denote the ring homomorphism defined as MapP , except that $\hat{c}_{5,x,i}^* \mapsto \tilde{r}'_{x,i}$ for every $(x, i) \in X_{0,\text{hon}} \times [L]$ (matching the entries $\langle \hat{c}_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$ recorded in the challenge phase of $\text{Hyb}_{5,0,\text{int}}^{(b)}$). By the same inductive argument as in Lemma 4.12, $\tilde{h} = \text{MapP}_{\text{int}}(\hat{h})$ for every handle $\langle h \rangle$ up to the first substitution point. From that point onwards, by the inductive argument of Lemma 4.15 (and since each $\text{SubS}_{X,Z}$ is a ring homomorphism applied to every entry), $\tilde{h}$ is obtained by applying the substitutions performed so far to $\text{MapP}_{\text{int}}(\hat{h})$. Hence, for any χ^{th} post-challenge zero-test query, the challenger's formal polynomial in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$ can be derived from the adversary's formal polynomial via the following substitutions (where $Z = Z_{\chi'}$):

$$\begin{aligned} \mathbb{G}_1 \begin{cases} \hat{y}_\zeta \mapsto \text{Corr}_Z(\tilde{y}_\zeta) & \text{for } \zeta \in [q_{\text{AS}}] \\ \hat{y}_{i,\zeta} \mapsto \text{Corr}_Z(\tilde{y}_{i,\zeta}) & \text{for } (i, \zeta) \in [L] \times [q_{\text{AS}}] \\ \hat{c}_{3,x}^* \mapsto \tilde{r}_x & \text{for } x \in [\ell] \end{cases} & \quad \hat{c}_{4,x}^* \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\text{T}} \tilde{\mathbf{w}} + \tilde{r}_x \tilde{y}_{a_x}) & \text{for } x \in [\ell] \\ \hat{c}_{5,x,i}^* \mapsto \begin{cases} \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,a_x}) & \text{if } i \in S_x^* \wedge \chi' \geq \chi \vee x \in X_{\chi',\text{cor}} \\ \tilde{r}'_{x,i} & \text{otherwise} \end{cases} & \text{for } (x, i) \in [\ell] \times [L], \end{aligned}$$

$$\mathbb{G}_2 \begin{cases} \hat{u}_j \mapsto \tilde{u}_j & \text{for } j \in [K] \\ \hat{\beta}_{i,\xi} \mapsto \tilde{\beta}_{i,\xi} & \text{for } (i, \xi) \in [L] \times [q_{\text{KG}}] \end{cases} \quad \hat{\delta}_\xi \mapsto \text{Corr}_Z\left(\tilde{\alpha}_\zeta + \tilde{y}_\zeta \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi}\right) \quad \text{for } \xi \in [q_{\text{KG}}],$$

$$\mathbb{G}_T \begin{cases} \hat{\alpha}_\zeta \mapsto \text{Corr}_Z(\tilde{\alpha}_\zeta) & \text{for } \zeta \in [q_{\text{AS}}] \\ \hat{c}_1^* \mapsto \tilde{s} + \tilde{m}_b^* \end{cases} \quad \hat{c}_{2,x}^* \mapsto \text{Corr}_Z(\mathbf{m}_x^{*\text{T}} \tilde{\mathbf{v}} + \tilde{r}_x \tilde{\alpha}_{a_x}) \quad \text{for } x \in [\ell], \quad (4.3)$$

where $\tilde{m}_b^*$, $\tilde{\alpha}_{a_x}$, $\tilde{y}_{a_x}$, and $\tilde{y}_{i,a_x}$ are as in Eq. (4.1). For the challenge policy $P^* = (\mathbf{M}^*, \rho^*)$, we write $\mathbf{m}_x^{*\text{T}} = (m_{x,1}, \dots, m_{x,n})$ for the x^{th} row of $\mathbf{M}^*$ and, recalling that $\tilde{\mathbf{v}} = (\tilde{s}, \tilde{v}_2, \dots, \tilde{v}_n)$ and $\tilde{\mathbf{w}} = (0, \tilde{w}_2, \dots, \tilde{w}_n)$, we have that $\mathbf{m}_x^{*\text{T}} \tilde{\mathbf{v}} = m_{x,1} \tilde{s} + \sum_{k \in [2,n]} m_{x,k} \tilde{v}_k$ and $\mathbf{m}_x^{*\text{T}} \tilde{\mathbf{w}} = \sum_{k \in [2,n]} m_{x,k} \tilde{w}_k$. We furthermore write $X_{\text{hon}} = X_{\chi,\text{hon}}$ for the set of rows x whose authority $a_x = \rho^*(x)$ is honest at the time of the χ^{th} post-challenge zero-test query (i.e., $a_x \in A$ and $\zeta_x \notin Z$ for the index ζ_x of the authority-setup query on a_x) and $X_{\text{cor}} = X_{\chi,\text{cor}}$ for the set of rows whose authority is corrupt. Note that $\tilde{\alpha}_{a_x}$, $\tilde{y}_{a_x}$, and $\tilde{y}_{i,a_x}$ are the variables $\tilde{\alpha}_{\zeta_x}$, $\tilde{y}_{\zeta_x}$, and $\tilde{y}_{i,\zeta_x}$ for $x \in X_{\text{hon}}$, while for $x \in X_{\text{cor}}$ they are polynomials associated with handles created before the challenge phase, in which no variable sampled during the challenge phase (e.g., $\tilde{s}$, $\tilde{r}_{x'}$, or $\tilde{r}'_{x',i}$) appears.

We are now ready to go over our main analysis of the χ^{th} zero-test query, which broadly breaks down into three cases. The first two analyze what happens when the adversary “combines” keys for fewer than or at least K distinct global identifiers respectively to break the system, while the third covers remaining edge cases. In more detail (but still at a high level), for an adversary to distinguish the two hybrids, it must query an adversary formal polynomial $\hat{h}_Z$ that maps to an identically zero challenger formal polynomial $\tilde{h}$ where $\hat{h}_Z$ has a $\hat{c}_1^*$ term and/or a $\hat{c}_{5,x,i}^*$ term for some

$(x, i) \in X_{\text{hon}} \times [L]$ and $\hat{v} \in \{1\} \cup \hat{\mathcal{V}}_2$, per the conditions added in the subsequent hybrid. Below, we provide intuition for which terms in $\hat{h}_Z$ are necessary to keep track of in our analysis (this applies to both the cases of combining keys for fewer than and at least K distinct global identifiers).

If $\hat{h}_Z$ has a $\hat{c}_{2,x}^*$ term, then by construction this induces a $\tilde{s}$ term in $\tilde{h}$, which the adversary can attempt to cancel out by including other terms in $\hat{h}_Z$ which induce $\tilde{s}$ terms in $\tilde{h}$. By inspection of Eq. (4.3) (and as we will justify in more detail later), $\tilde{s}$ can only otherwise be obtained from the $\hat{c}_{2,x}^*$ terms (i.e., $\hat{h}_Z$ must include $\hat{c}_{2,x}^*$ terms for any cancellation to occur). Since these same terms also induce the $\tilde{v}_k$ terms for $k \in [2, n]$, which must themselves cancel, the coefficients $\text{coeff}_{\hat{c}_{2,x}^*}(\hat{h}_Z)$ must form a reconstruction vector for the challenge policy P^* (in particular, recall that $\tilde{v}$ shares $\tilde{s}$).

Note that, for a row $x \in X_{\text{hon}}$, the $\hat{c}_{2,x}^*$ term in $\hat{h}_Z$ also induces a $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ term in $\tilde{h}$, and this can only be cancelled by including *additional* terms in $\hat{h}_Z$ which induce $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ terms in $\tilde{h}$. As shown in Eq. (4.3) (and as we will again justify in more detail a bit later), the only other way to obtain $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ is to pair $\hat{c}_{3,x}^*$ with some $\hat{\delta}_\xi$ for a key ξ issued on the attribute a_x (i.e., $\hat{h}_Z$ must include $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms for any cancellation to occur).

In turn, the $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms induce both $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$ terms and $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$. The former can only otherwise be induced by $\hat{c}_{4,x}^* \hat{u}_j$ terms, which themselves induce $\tilde{w}_k \tilde{u}_j$ terms. Informally, these $\tilde{w}_k \tilde{u}_j$ terms cancel only if all rows use keys issued on the same global identifier gid (since $\tilde{w}$ shares 0), which would mean a *single* global identifier satisfies P^* , contradicting admissibility. The latter can only otherwise be induced by $\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}$ terms, and only when $i \in S_x^*$, which is what forces $S_\xi \subseteq S_x^*$.

For any remaining terms of the form $\hat{c}_{5,x,i}^* \hat{v}$ for $\hat{v} \in \{1\} \cup \hat{\mathcal{V}}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]}$ (i.e., the other challenge-set-dependent terms), a simple linear independence argument shows that cancellation does not occur if they are present.

To formalize the above intuition, consider the following events based on the adversary's formal polynomial $\hat{h}_Z$:

$$\begin{aligned} E_1 &:= [\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \neq 0], \\ E_2 &:= [\text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \neq 0 \text{ for some } (x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}]], \\ E_3 &:= [\text{coeff}_{\hat{c}_{5,x,i}^* \hat{v}}(\hat{h}_Z) \neq 0 \text{ for some } (x, i, \hat{v}) \in X_{\text{hon}} \times [L] \times (\{1\} \cup \hat{\mathcal{V}}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]})]. \end{aligned}$$

By construction, we have that E_1 is the message-dependent condition and $E_2 \vee E_3$ is the challenge-set-dependent condition, and thus the bad event can only occur in events E_1, E_2 , or E_3 . Next, define

$$D = \{(x, \xi) \in X_{\text{hon}} \times [q_{\text{KG}}] : a_\xi = a_x \wedge \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z) \neq 0\} \quad \text{and} \quad G = \{\text{gid}_\xi : (x, \xi) \in D\},$$

i.e., D is the set of indices for the $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms in $\hat{h}_Z$ with an honest row x and a key ξ issued on its attribute a_x and G is the set of distinct global identifiers of the keys they involve, and consider the following event that captures when the adversary “combines” keys for fewer than K distinct global identifiers:

$$E_4 := [|G| < K].$$

With the above events, we can bound the probability of the bad event as

$$\begin{aligned} \Pr[\text{bad}] &= \Pr[\text{bad} \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge (E_1 \vee E_2 \vee E_3) \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &= \Pr[\text{bad} \wedge (E_1 \vee (\neg E_1 \wedge E_2) \vee E_3) \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge E_1 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4] + \Pr[\text{bad} \wedge E_3 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge E_1 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4] + \Pr[\text{bad} \wedge E_3] + \Pr[\text{bad} \wedge \neg E_4] \\ &= \Pr[\text{bad}_{\text{I-a}}] + \Pr[\text{bad}_{\text{I-b}}] + \Pr[\text{bad}_{\text{III}}] + \Pr[\text{bad}_{\text{II}}], \end{aligned}$$

where

$$\begin{aligned} \text{bad}_{\text{I-a}} &:= \text{bad} \wedge E_1 \wedge E_4 & \text{bad}_{\text{I-b}} &:= \text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4 \\ \text{bad}_{\text{II}} &:= \text{bad} \wedge \neg E_4 & \text{bad}_{\text{III}} &:= \text{bad} \wedge E_3. \end{aligned}$$

We now analyze the individual bad sub-cases below.

- **Case I: $\hat{h}_Z$ combines keys for fewer than K distinct global identifiers:** We begin by analyzing the two bad events in which $|G| < K$. Before doing so, recall that $\hat{h}_Z = \text{Expd}_{Z_X}(\hat{\mathcal{T}}(\langle h \rangle))$, where Expd_{Z_X} replaces each $\hat{\delta}_\xi$ for a key ξ issued on a corrupt attribute with its expansion in terms of variables of the form $\hat{u}_j$ and $\hat{\beta}_{i,\xi}$. We will leverage two properties of this expansion (which also apply to Cases II and III). Firstly, $\hat{h}_Z$ contains no $\hat{\delta}_\xi$ for a key ξ issued on a corrupt attribute. Second, the expansion does not affect the challenger's formal polynomial $\tilde{h}$. In particular, if the authority $\zeta = \zeta(\xi)$ associated with a_ξ is corrupt, then Eq. (4.3) maps $\hat{\delta}_\xi$ to $\alpha_\zeta + y_\zeta \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \gamma_{i,\zeta} \tilde{\beta}_{i,\xi}$ (because Corr_Z substitutes in $\alpha_\zeta, y_\zeta, \gamma_{i,\zeta}$), which is what $\text{Expd}_{Z_X}(\hat{\delta}_\xi)$ produces in this case. In particular, every key ξ with $(x, \xi) \in D$ is issued on an honest attribute. Continuing, as mentioned before we will analyze the following two sub-cases.

- **Case I-a: $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \neq 0$:** At a high level, we analyze this case by arguing that a $\hat{c}_1^*$ term in $\hat{h}_Z$ induces an $\tilde{s}$ term in $\tilde{h}$, which can only be cancelled by $\hat{c}_{2,x}^*$ terms in $\hat{h}_Z$ (as these are the only other terms that also induce $\tilde{s}$ terms in $\tilde{h}$) whose coefficients reconstruct the challenge policy P^* . For honest rows x , these $\hat{c}_{2,x}^*$ terms also induce $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ terms in $\tilde{h}$, which can only be cancelled by $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms, whose $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$ terms can in turn only be cancelled by $\hat{c}_{4,x}^* \hat{u}_j$ terms, which themselves induce $\tilde{w}_k \tilde{u}_j$ terms. If keys for fewer than K distinct global identifiers are combined, a cancellation cannot occur between all of these terms unless the coefficients associated with a *single* global identifier already reconstruct P^* , which the admissibility of $\mathcal{A}$ precludes.

We now identify which terms (that are not $\hat{c}_1^*$) in $\hat{h}_Z$ can induce $\tilde{s}$ or $\tilde{v}_k$ terms in $\tilde{h}$. From Eq. (4.3), we can see that no variable in $\hat{\mathcal{V}}_1$ or $\hat{\mathcal{V}}_2$ induces the variable $\tilde{s}$ or $\tilde{v}_k$, and so neither can any pairing of a variable in $\hat{\mathcal{V}}_1$ with a variable in $\hat{\mathcal{V}}_2$. Thus, the only other type of term in $\hat{h}_Z$ that can induce them is $\hat{c}_{2,x}^*$ for $x \in [\ell]$, which specifically induces $\mathbf{m}_x^T \tilde{\mathbf{v}} = m_{x,1} \tilde{s} + \sum_{k \in [2,n]} m_{x,k} \tilde{v}_k$.

Note that, for a row $x \in X_{\text{hon}}$, $\hat{c}_{2,x}^*$ also induces an $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ term in $\tilde{h}$ in addition to $\tilde{s}$ and $\tilde{v}_k$ terms. As these, along with the $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$ and $\tilde{w}_k \tilde{u}_j$ terms that are needed to cancel them, are the terms we will use to argue that no cancellation can occur when keys for fewer than K distinct global identifiers are combined, we also identify which other terms in $\hat{h}_Z$ can induce $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$, $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$, or $\tilde{w}_k \tilde{u}_j$ terms in $\tilde{h}$. Besides $\hat{c}_{2,x}^*$ itself, a term inducing $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ must be a pairing of a variable in $\hat{\mathcal{V}}_1$ inducing a (solo) $\tilde{r}_x$ with a variable in $\hat{\mathcal{V}}_2$ inducing a (solo) $\tilde{\alpha}_{\zeta_x}$. From Eq. (4.3), we can see that $\hat{c}_{3,x}^*$ is the only variable in $\hat{\mathcal{V}}_1$ that satisfies the former and that $\hat{\delta}_\xi$ for a key ξ issued on the attribute a_x is the only type of variable in $\hat{\mathcal{V}}_2$ that satisfies the latter. Next, a term inducing $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$ must be a pairing of a variable in $\hat{\mathcal{V}}_1$ inducing $\tilde{r}_x$ or $\tilde{r}_x \tilde{y}_{\zeta_x}$ with a variable in $\hat{\mathcal{V}}_2$ inducing $\tilde{y}_{\zeta_x} \tilde{u}_j$ or $\tilde{u}_j$ respectively, and from Eq. (4.3), the only options are to pair $\hat{c}_{3,x}^*$ with $\hat{\delta}_\xi$ for a key ξ issued on a_x , or to pair $\hat{c}_{4,x}^*$ with $\hat{u}_j$. Finally, $\hat{c}_{4,x'}^*$ for $x' \in [\ell]$ is the only type of variable in $\hat{\mathcal{V}}_1$ inducing a $\tilde{w}_k$ variable and (as $\hat{h}_Z$ contains no $\hat{\delta}_\xi$ for keys issued on corrupt attributes) $\hat{u}_j$ is the only variable in $\hat{\mathcal{V}}_2$ inducing a (solo) $\tilde{u}_j$ variable, so a term inducing $\tilde{w}_k \tilde{u}_j$ must be $\hat{c}_{4,x'}^* \hat{u}_j$ for some $x' \in [\ell]$. Altogether, the only terms in $\hat{h}_Z$ that induce $\tilde{s}$, $\tilde{v}_k$, $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$, $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$, or $\tilde{w}_k \tilde{u}_j$ terms in $\tilde{h}$ (for $x \in X_{\text{hon}}$) are $\hat{c}_1^*$, $\hat{c}_{2,x}^*$, $\hat{c}_{3,x}^* \hat{\delta}_\xi$ for keys ξ issued on a_x , and $\hat{c}_{4,x}^* \hat{u}_j$.

Recalling that $D = \{(x, \xi) \in X_{\text{hon}} \times [q_{KG}] : a_\xi = a_x \wedge \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z) \neq 0\}$, we can write the challenger's formal polynomial $\tilde{h}$ corresponding to the adversary's formal polynomial $\hat{h}_Z$ on the χ^{th} query as

$$\begin{aligned}
\tilde{h} = & \tilde{\mu} + \text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \cdot \tilde{s} + \sum_{x \in [\ell]} \text{coeff}_{\hat{c}_{2,x}^*}(\hat{h}_Z) \cdot \mathbf{m}_x^T \tilde{\mathbf{v}} + \sum_{x \in X_{\text{hon}}} \text{coeff}_{\hat{c}_{2,x}^*}(\hat{h}_Z) \cdot \tilde{r}_x \tilde{\alpha}_{\zeta_x} \\
& + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \left(\tilde{\alpha}_{\zeta_x} + \tilde{y}_{\zeta_x} \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{\gamma}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \right) \\
& + \sum_{(x,j) \in [\ell] \times [K]} \text{coeff}_{\hat{c}_{4,x}^* \hat{u}_j}(\hat{h}_Z) \cdot \mathbf{m}_x^T \tilde{\mathbf{w}} \tilde{u}_j + \sum_{(x,j) \in X_{\text{hon}} \times [K]} \text{coeff}_{\hat{c}_{4,x}^* \hat{u}_j}(\hat{h}_Z) \cdot \tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j,
\end{aligned} \tag{4.4}$$

where $\tilde{\mu}$ is a polynomial over $\tilde{\mathcal{V}}$ such that $\text{coeff}_{\tilde{s}}(\tilde{\mu}) = 0$, $\text{coeff}_{\tilde{v}_k}(\tilde{\mu}) = 0$, and $\text{coeff}_{\tilde{w}_k \tilde{u}_j}(\tilde{\mu}) = 0$ for all $(k, j) \in [2, n] \times [K]$, and $\text{coeff}_{\tilde{r}_x \tilde{\alpha}_{\zeta_x}}(\tilde{\mu}) = 0$ and $\text{coeff}_{\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j}(\tilde{\mu}) = 0$ for all $(x, j) \in X_{\text{hon}} \times [K]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{s}$, $\tilde{v}_k$, $\tilde{w}_k \tilde{u}_j$, $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$, or $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$ terms). This holds because, as argued above, $\hat{c}_1^*$, $\hat{c}_{2,x}^*$, $\hat{c}_{3,x}^* \hat{\delta}_{\xi}$, and $\hat{c}_{4,x}^* \hat{u}_j$ are the only terms in $\hat{h}_Z$ that induce these terms in $\tilde{h}$, and because the remaining terms induced by $\hat{c}_{2,x}^*$ and $\hat{c}_{4,x}^* \hat{u}_j$ for $x \in X_{\text{cor}}$ (namely, $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ and $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$), which are absorbed into $\tilde{\mu}$, contain none of these terms as $\tilde{\alpha}_{\zeta_x}$ and $\tilde{y}_{\zeta_x}$ contain no variables sampled during the challenge phase. Note that while $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \cdot \tilde{m}_b^*$ is absorbed into $\tilde{\mu}$, the above holds because $\tilde{m}_b^*$ contains none of these terms, due to the fact that $\langle m_b^* \rangle$ must have been created before the challenge phase and none of $\tilde{s}$, $\tilde{v}_k$, $\tilde{w}_k$, or $\tilde{r}_x$ appears in any $\tilde{\mathcal{T}}$ entry before then.

Next, define the vector

$$\boldsymbol{\theta} = (\theta_1, \dots, \theta_{\ell}) = (\text{coeff}_{\hat{c}_{2,1}^*}(\hat{h}_Z), \dots, \text{coeff}_{\hat{c}_{2,\ell}^*}(\hat{h}_Z)),$$

and for each $j \in [K]$ define the vector

$$\boldsymbol{\eta}_j = (\eta_{1,j}, \dots, \eta_{\ell,j}) = (\text{coeff}_{\hat{c}_{4,1}^* \hat{u}_j}(\hat{h}_Z), \dots, \text{coeff}_{\hat{c}_{4,\ell}^* \hat{u}_j}(\hat{h}_Z)),$$

and finally, for each $g \in G$, define the vector $\boldsymbol{\psi}_g = (\psi_{1,g}, \dots, \psi_{\ell,g})$ with entries

$$\psi_{x,g} = \sum_{\xi \in [q_{KG}] : (x,\xi) \in D \wedge \text{gid}_{\xi} = g} \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_{\xi}}(\hat{h}_Z).$$

Note that $\psi_{x,g} = 0$ for every $x \in X_{\text{cor}}$, as $D \subseteq X_{\text{hon}} \times [q_{KG}]$. Recalling that $\mathbf{m}_x^{*\top} \tilde{\mathbf{v}} = m_{x,1} \tilde{s} + \sum_{k \in [2,n]} m_{x,k} \tilde{v}_k$ and $\mathbf{m}_x^{*\top} \tilde{\mathbf{w}} = \sum_{k \in [2,n]} m_{x,k} \tilde{w}_k$ and rearranging Eq. (4.4) further, we have that

$$\begin{aligned} \tilde{h} &= \tilde{\mu}' + \tilde{s} \cdot \left(\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) + \sum_{x \in [\ell]} \theta_x m_{x,1} \right) + \sum_{k \in [2,n]} \tilde{v}_k \cdot \left(\sum_{x \in [\ell]} \theta_x m_{x,k} \right) + \sum_{x \in X_{\text{hon}}} \tilde{r}_x \tilde{\alpha}_{\zeta_x} \cdot \left(\theta_x + \sum_{g \in G} \psi_{x,g} \right) \\ &\quad + \sum_{(x,j) \in X_{\text{hon}} \times [K]} \tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j \cdot \left(\eta_{x,j} + \sum_{g \in G} \psi_{x,g} \cdot g^{j-1} \right) + \sum_{(k,j) \in [2,n] \times [K]} \tilde{w}_k \tilde{u}_j \cdot \left(\sum_{x \in [\ell]} \eta_{x,j} m_{x,k} \right) \\ &= \tilde{\mu}' + \tilde{s} \cdot \left(\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) + \boldsymbol{\theta}^\top \mathbf{M}^* \mathbf{e}_1 \right) + \sum_{k \in [2,n]} \tilde{v}_k \cdot \left(\boldsymbol{\theta}^\top \mathbf{M}^* \mathbf{e}_k \right) + \sum_{x \in X_{\text{hon}}} \tilde{r}_x \tilde{\alpha}_{\zeta_x} \cdot \left(\boldsymbol{\theta} + \sum_{g \in G} \boldsymbol{\psi}_g \right)_x \\ &\quad + \sum_{(x,j) \in X_{\text{hon}} \times [K]} \tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j \cdot \left(\boldsymbol{\eta}_j + \sum_{g \in G} g^{j-1} \boldsymbol{\psi}_g \right)_x + \sum_{(k,j) \in [2,n] \times [K]} \tilde{w}_k \tilde{u}_j \cdot \left(\boldsymbol{\eta}_j^\top \mathbf{M}^* \mathbf{e}_k \right) \end{aligned} \quad (4.5)$$

where $\tilde{\mu}' = \tilde{\mu} + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_{\xi}}(\hat{h}_Z) \cdot \sum_{i \in S_{\xi}} \tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi}$. Intuitively, we have absorbed the $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi}$ terms into $\tilde{\mu}'$ so that $\tilde{\mu}'$ contains all the terms that are not core to the analysis of the $|G| < K$ case. By construction, we have that $\tilde{\mu}'$ does not contain any $\tilde{s}$, $\tilde{v}_k$, $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$, $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$, or $\tilde{w}_k \tilde{u}_j$ terms.

Observe that if any of the coefficients in parentheses in Eq. (4.5) is nonzero, then $\tilde{h} \neq 0$. This is because each of these is the coefficient of a distinct $\tilde{s}$, $\tilde{v}_k$, $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$, $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$, or $\tilde{w}_k \tilde{u}_j$ term in $\tilde{h}$, while $\tilde{\mu}'$ does not contain any of these terms as shown above (i.e., there are no terms in $\tilde{\mu}'$ that can cancel them out).

Now suppose that all of the coefficients in parentheses in Eq. (4.5) are zero. Let $V_{\text{cor}} = \text{span}\{\mathbf{m}_x^{*\top}\}_{x \in X_{\text{cor}}}$ be the span of the rows of $\mathbf{M}^*$ that are labeled by corrupt attributes. We have that the following four equations hold. The first equation is that

$$\boldsymbol{\theta}^\top \mathbf{M}^* = -\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \cdot \mathbf{e}_1^\top, \quad (4.6a)$$

which follows because the $\tilde{s}$ and $\tilde{v}_k$ coefficients in Eq. (4.5) imply that $\theta^\top \mathbf{M}^* \mathbf{e}_1 = -\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z)$ and $\theta^\top \mathbf{M}^* \mathbf{e}_k = 0$ for every $k \in [2, n]$. The second equation is that

$$\left(\theta + \sum_{g \in G} \psi_g \right)^\top \mathbf{M}^* \in V_{\text{cor}}, \quad (4.6b)$$

which follows because the $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ coefficients state that the vector in parentheses has a zero entry for every row $x \in X_{\text{hon}}$, so that multiplying it by $\mathbf{M}^*$ only yields a linear combination of the rows labeled by corrupt attributes. The third equation is that

$$\left(\eta_j + \sum_{g \in G} g^{j-1} \psi_g \right)^\top \mathbf{M}^* \in V_{\text{cor}} \quad \text{for } j \in [K], \quad (4.7a)$$

which follows in the same way from the $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$ coefficients, as they state that the vector in parentheses has a zero entry for every row $x \in X_{\text{hon}}$. Finally, the fourth equation is that

$$\eta_j^\top \mathbf{M}^* \in \text{span}(\mathbf{e}_1^\top) \quad \text{for } j \in [K], \quad (4.7b)$$

which follows because the $\tilde{w}_k \tilde{u}_j$ coefficients state that $\eta_j^\top \mathbf{M}^* \mathbf{e}_k = 0$ for every $k \in [2, n]$. Subtracting the second equation from the first in Eq. (4.7) gives that

$$\sum_{g \in G} g^{j-1} \cdot \psi_g^\top \mathbf{M}^* \in V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top) \quad \text{for } j \in [K],$$

where $V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top)$ denotes the span of the rows of $\mathbf{M}^*$ labeled by corrupt attributes together with $\mathbf{e}_1^\top$.

Label the elements of $G = \{g_1, \dots, g_{|G|}\}$, let $\mathbf{V} = V(g_1, \dots, g_{|G|})$ be a square Vandermonde matrix of dimension $|G|$ (see Section 2), and let

$$\Psi = (\mathbf{M}^{*\top} \psi_{g_1}, \dots, \mathbf{M}^{*\top} \psi_{g_{|G|}})^\top,$$

i.e., the matrix whose i^{th} row is $\psi_{g_i}^\top \mathbf{M}^*$ for $i \in [|G|]$. As the j^{th} row of $\mathbf{V}\Psi$ is $\sum_{i \in [|G|]} g_i^{j-1} \cdot \psi_{g_i}^\top \mathbf{M}^*$, the above for $j \in [|G|]$ (which is well defined because $|G| < K$) states that every row of $\mathbf{V}\Psi$ lies in $V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top)$. Because the global identifiers $g_1, \dots, g_{|G|}$ are pair-wise distinct, we have that $\det(\mathbf{V}) \neq 0$ (i.e., $\mathbf{V}$ is invertible), and so $\Psi = \mathbf{V}^{-1}(\mathbf{V}\Psi)$, meaning that every row of Ψ is a linear combination of the rows of $\mathbf{V}\Psi$. Since $V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top)$ is closed under linear combination, it follows that $\psi_g^\top \mathbf{M}^* \in V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top)$ for every $g \in G$. Equivalently, for each $g \in G$ there exists a scalar $c_g \in \mathbb{Z}_p$ such that

$$\psi_g^\top \mathbf{M}^* - c_g \cdot \mathbf{e}_1^\top \in V_{\text{cor}}. \quad (4.8)$$

Summing Eq. (4.8) over $g \in G$ and substituting Eq. (4.6), we obtain

$$\left(\sum_{g \in G} c_g - \text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \right) \cdot \mathbf{e}_1^\top \in V_{\text{cor}}.$$

To conclude, recall that $\mathcal{A}$ is admissible and thus the attributes corrupted by the time of the χ^{th} post-challenge zero-test query remain corrupt for the rest of the experiment and the corrupt attributes on their own do not satisfy the challenge policy P^* (i.e., $\mathbf{e}_1^\top \notin V_{\text{cor}}$). It follows that $\sum_{g \in G} c_g = \text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \neq 0$.

In particular, there exists some $g \in G$ with $c_g \neq 0$. For this g , it follows from Eq. (4.8) that $\mathbf{e}_1^\top \in c_g^{-1} \psi_g^\top \mathbf{M}^* + V_{\text{cor}}$ (i.e., $\mathbf{e}_1^\top$ is a linear combination of the rows of $\mathbf{M}^*$ labeled by corrupt attributes and the rows $x \in X_{\text{hon}}$ with $\psi_{x,g} \neq 0$). By the definition of $\psi_{x,g}$, every such row $x \in X_{\text{hon}}$ is labeled by an attribute a_x on which $\mathcal{A}$ made a key-generation query with global identifier g , and so the corrupt attributes together with A_g satisfy P^* , contradicting the admissibility of $\mathcal{A}$. It follows that $\tilde{h} \neq 0$ and hence the zero-test oracle always returns “non-zero” in this case. Thus,

$$\Pr[\text{bad}_{\Gamma-a}] = 0.$$

- **Case I-b: $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) = 0$ and $\text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \neq 0$ for some $(x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}]$:** At a very high level, we analyze this case by arguing that a $\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}$ term in $\hat{h}_Z$ induces $\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$. The former can only be cancelled by including $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms in $\hat{h}_Z$ (as these are the only other terms that also induce $\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi}$), and then only when $i = j$ and the key ξ is issued on the attribute a_x . If keys for fewer than K distinct global identifiers are combined, such a cancellation cannot occur per an argument similar to Case I-a together with the linear independence of the challenge policy for unauthorized sets. For the latter, the lack of cancellation is straightforward but tedious.

We now identify which terms in $\hat{h}_Z$ can induce terms of the form $\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}$ in $\tilde{h}$. By inspecting Eq. (4.3), we can see that there are two types of terms in $\hat{h}_Z$ that can induce the formal variable $\tilde{\beta}_{j,\xi}$ in $\tilde{h}$. These are $\hat{\beta}_{j,\xi}$ and $\hat{\delta}_\xi$ which are both in $\mathcal{V}_2$. For the former, it must be paired with some $\hat{v} \in \mathcal{V}_1$ that induces terms of the form $\tilde{r}_x \tilde{y}_{i,\xi_x}$ or $\tilde{r}'_{x,i}$, and from Eq. (4.3) we can see that $\hat{c}_{5,x,i}^*$ is the only type of variable in $\mathcal{V}_1$ that satisfies this. For the latter, it must be paired with some $\hat{v} \in \mathcal{V}_1$ that induces a solo $\tilde{r}_x$ variable (i.e., without any additional variables multiplied), and from Eq. (4.3) we can see only $\hat{c}_{3,x}^*$ satisfies this; moreover, $\hat{c}_{3,x}^* \hat{\delta}_\xi$ induces $\tilde{r}_x \tilde{y}_{j,\xi} \tilde{\beta}_{j,\xi}$ only for the authority $\zeta = \zeta(\xi)$ on which the key ξ was issued and only for $j \in S_\xi$.

Let $C = \{(x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}] : \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \neq 0\}$ be the set of indices for $\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}_Z$ and define the following sets:

$$\begin{aligned} C' &= \{(x, i, \xi) : (x, i, j, \xi) \in C \wedge i = j\}, & C'' &= \{(x, i, j, \xi) \in C : i \neq j\}, \\ C_{S^*}' &= \{(x, i, \xi) \in C' : i \in S_x^*\}, & C_{S^*}'' &= \{(x, i, j, \xi) \in C'' : i \in S_x^*\}. \end{aligned}$$

Intuitively, C' and C'' can be thought of as partitioning C into the “diagonal” and “off-diagonal” $\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}_Z$. Furthermore, C_{S^*}' and C_{S^*}'' are the subsets that induce $\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$ (because $i \in S_x^*$) as opposed to $\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$.

Like Case I-a, recall that $D = \{(x, \xi) \in X_{\text{hon}} \times [q_{\text{KG}}] : a_\xi = a_x \wedge \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z) \neq 0\}$. We can write the challenger’s formal polynomial $\tilde{h}$ corresponding to the adversary’s formal polynomial $\hat{h}_Z$ on the χ^{th} query as

$$\begin{aligned} \tilde{h} &= \tilde{\mu} + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \left(\tilde{\alpha}_{\xi_x} + \tilde{y}_{\xi_x} \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\xi_x} \tilde{\beta}_{i,\xi} \right) \\ &+ \sum_{(x,i,\xi) \in C_{S^*}'} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{i,\xi} \right) + \sum_{(x,i,\xi) \in C' \setminus C_{S^*}'} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}'_{x,i} \tilde{\beta}_{i,\xi} \right) \\ &+ \sum_{(x,i,j,\xi) \in C_{S^*}''} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi} \right) + \sum_{(x,i,j,\xi) \in C'' \setminus C_{S^*}''} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi} \right) \end{aligned} \quad (4.9)$$

where $\tilde{\mu}$ is a polynomial over $\mathcal{V}$ such that $\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = \text{coeff}_{\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = 0$ for all $(x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}$ terms). This holds because, as argued above, $\hat{c}_{3,x}^* \hat{\delta}_\xi$ and $\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}$ are the only terms in $\hat{h}_Z$ that can induce $\tilde{r}_x \tilde{y}_{i,\xi_x} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}$ terms (where the $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms for keys ξ not issued on a_x , which are absorbed into $\tilde{\mu}$, only induce such terms with $\tilde{y}_{i,\xi}$ for $\zeta \neq \zeta_x$), and we do not need to consider the $\tilde{m}_b^*$ term induced by $\hat{c}_1^*$ because we assume that $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) = 0$ in this case.

By inspection of Eq. (4.9) and using the above fact on $\tilde{\mu}$, we can see that

$$\begin{aligned} \text{coeff}_{\tilde{r}'_{x,i}\tilde{\beta}_{i,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{i,\xi}}(\hat{h}_Z) && \text{for } (x, i, \xi) \in C' \setminus C'_{S^*}, \\ \text{coeff}_{\tilde{r}_x\tilde{y}_{i,\xi}\tilde{\beta}_{i,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{i,\xi}}(\hat{h}_Z) && \text{for } (x, i, j, \xi) \in C''_{S^*}, \\ \text{coeff}_{\tilde{r}'_{x,i}\tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{j,\xi}}(\hat{h}_Z) && \text{for } (x, i, j, \xi) \in C'' \setminus C''_{S^*}, \end{aligned}$$

(i.e., the only cancellation that can occur is on $\hat{c}_{5,x,i}\hat{\beta}_{i,\xi}$ terms in $\hat{h}_Z$ that induce $\tilde{r}_x\tilde{y}_{i,\xi}\tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$). Thus, if C'_{S^*} , $C'' \setminus C''_{S^*}$, or $C' \setminus C'_{S^*}$ are non-empty, then $\tilde{h} \neq 0$.

Now suppose that C'_{S^*} , $C'' \setminus C''_{S^*}$, and $C' \setminus C'_{S^*}$ are all empty (note that this implies that C'_{S^*} must be non-empty, as we assume $\text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{j,\xi}}(\hat{h}_Z) \neq 0$ for some $(x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}]$) in this case. We can simplify Eq. (4.9) to

$$\begin{aligned} \tilde{h} &= \tilde{\mu} + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}\hat{\delta}_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \left(\tilde{\alpha}_{\zeta_x} + \tilde{y}_{\zeta_x} \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \right) \\ &\quad + \sum_{(x,i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \right). \end{aligned} \quad (4.10)$$

Rearranging Eq. (4.10) further, we have that

$$\begin{aligned} \tilde{h} &= \tilde{\mu} + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}\hat{\delta}_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \left(\tilde{\alpha}_{\zeta_x} + \tilde{y}_{\zeta_x} \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j \right) \\ &\quad + \sum_{(x,i,\xi) \in U \cap C'_{S^*}} \tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_{3,x}\hat{\delta}_\xi}(\hat{h}_Z) + \text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{i,\xi}}(\hat{h}_Z) \right) \\ &\quad + \sum_{(x,i,\xi) \in U \setminus C'_{S^*}} \text{coeff}_{\hat{c}_{3,x}\hat{\delta}_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} + \sum_{(x,i,\xi) \in C'_{S^*} \setminus U} \text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \end{aligned} \quad (4.11)$$

where $U = \bigcup_{(x,\xi) \in D} \{(x, i, \xi) : i \in S_\xi\}$ is the set of indices (x, i, ξ) such that $(x, \xi) \in D$ and $i \in S_\xi$, i.e., the indices that appear in the left-hand sum in Eq. (4.10).

For the remainder, we will argue that if D is empty, then Eq. (4.11) shows that $\tilde{h}$ is not identically zero; while, on the other hand, if D is non-empty, then we can apply the analysis of Case I-a together with the linear independence of the challenge policy for unauthorized sets to show that $\tilde{h}$ is again not identically zero.

Now suppose that D is empty. By inspection of Eq. (4.11) and using the fact C'_{S^*} is non-empty (as argued above), there exists some $(x, i, \xi) \in C'_{S^*} = C'_{S^*} \setminus U$ such that

$$\text{coeff}_{\tilde{r}_x\tilde{y}_{i,\zeta_x}\tilde{\beta}_{i,\xi}}(\tilde{h}) = \text{coeff}_{\hat{c}_{5,x,i}\hat{\beta}_{i,\xi}}(\hat{h}_Z) \neq 0,$$

and so $\tilde{h} \neq 0$ in this case.

Now suppose D is non-empty. As in Case I-a, define the vector

$$\theta = (\theta_1, \dots, \theta_\ell) = (\text{coeff}_{\hat{c}_{2,1}}(\hat{h}_Z), \dots, \text{coeff}_{\hat{c}_{2,\ell}}(\hat{h}_Z)),$$

and, for each $j \in [K]$, the vector

$$\eta_j = (\eta_{1,j}, \dots, \eta_{\ell,j}) = (\text{coeff}_{\hat{c}_{4,1}\hat{u}_j}(\hat{h}_Z), \dots, \text{coeff}_{\hat{c}_{4,\ell}\hat{u}_j}(\hat{h}_Z))$$

and finally the vector $\psi_g = (\psi_{1,g}, \dots, \psi_{\ell,g})$ for each $g \in G$ with entries

$$\psi_{x,g} = \sum_{\xi \in [q_{KG}] : (x,\xi) \in D \wedge \text{gid}_\xi = g} \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z),$$

where again $\psi_{x,g} = 0$ for every $x \in X_{\text{cor}}$. Because $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) = 0$ in this case, the same reasoning as in Case I-a gives that

$$\begin{aligned} \tilde{h} = & \tilde{\mu}' + \tilde{s} \cdot \left(\theta^\top \mathbf{M}^* \mathbf{e}_1 \right) + \sum_{k \in [2,n]} \tilde{v}_k \cdot \left(\theta^\top \mathbf{M}^* \mathbf{e}_k \right) + \sum_{x \in X_{\text{hon}}} \tilde{r}_x \tilde{\alpha}_{\zeta_x} \cdot \left(\theta + \sum_{g \in G} \psi_g \right)_x \\ & + \sum_{(x,j) \in X_{\text{hon}} \times [K]} \tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j \cdot \left(\eta_j + \sum_{g \in G} g^{j-1} \psi_g \right)_x + \sum_{(k,j) \in [2,n] \times [K]} \tilde{w}_k \tilde{u}_j \cdot \left(\eta_j^\top \mathbf{M}^* \mathbf{e}_k \right), \end{aligned} \quad (4.12)$$

where $\tilde{\mu}'$ does not contain any $\tilde{s}$, $\tilde{v}_k$, $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$, $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$, or $\tilde{w}_k \tilde{u}_j$ terms. Thus, if any of the coefficients in parentheses in Eq. (4.12) is nonzero, then $\tilde{h} \neq 0$, as each is the coefficient of a distinct term in $\tilde{h}$ that $\tilde{\mu}'$ cannot cancel out.

So suppose for the sake of contradiction that all of the coefficients in parentheses in Eq. (4.12) are zero, and let $V_{\text{cor}} = \text{span}\{\mathbf{m}_x^*\}_{x \in X_{\text{cor}}}$. Then

$$\left(\eta_j + \sum_{g \in G} g^{j-1} \psi_g \right)^\top \mathbf{M}^* \in V_{\text{cor}} \quad \text{and} \quad \eta_j^\top \mathbf{M}^* \in \text{span}(\mathbf{e}_1^\top) \quad \text{for } j \in [K],$$

where the former follows because the $\tilde{r}_x \tilde{y}_{\zeta_x} \tilde{u}_j$ coefficients state that the vector in parentheses has a zero entry for every row $x \in X_{\text{hon}}$, and the latter follows because the $\tilde{w}_k \tilde{u}_j$ coefficients state that $\eta_j^\top \mathbf{M}^* \mathbf{e}_k = 0$ for every $k \in [2, n]$. Subtracting the latter from the former gives that

$$\sum_{g \in G} g^{j-1} \cdot \psi_g^\top \mathbf{M}^* \in V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top) \quad \text{for } j \in [K].$$

Label the elements of $G = \{g_1, \dots, g_{|G|}\}$, let $\mathbf{V} = V(g_1, \dots, g_{|G|})$ be a square Vandermonde matrix of dimension $|G|$, and let $\Psi = (\mathbf{M}^{*\top} \psi_{g_1}, \dots, \mathbf{M}^{*\top} \psi_{g_{|G|}})^\top$. As the j^{th} row of $\mathbf{V}\Psi$ is $\sum_{i \in [|G|]} g_i^{j-1} \cdot \psi_{g_i}^\top \mathbf{M}^*$, the above for $j \in [|G|]$ (which is well defined because $|G| < K$) states that every row of $\mathbf{V}\Psi$ lies in $V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top)$. Because $g_1, \dots, g_{|G|}$ are pair-wise distinct, $\mathbf{V}$ is invertible, and so every row of $\Psi = \mathbf{V}^{-1}(\mathbf{V}\Psi)$ also lies in $V_{\text{cor}} + \text{span}(\mathbf{e}_1^\top)$. That is, for each $g \in G$ there exists a scalar $c_g \in \mathbb{Z}_p$ such that

$$\psi_g^\top \mathbf{M}^* - c_g \cdot \mathbf{e}_1^\top \in V_{\text{cor}}. \quad (4.13)$$

If $c_g \neq 0$ for some $g \in G$, then Eq. (4.13) states that $\mathbf{e}_1^\top \in c_g^{-1} \psi_g^\top \mathbf{M}^* + V_{\text{cor}}$ (i.e., $\mathbf{e}_1^\top$ is a linear combination of the rows of $\mathbf{M}^*$ labeled by corrupt attributes and the rows $x \in X_{\text{hon}}$ with $\psi_{x,g} \neq 0$). By the definition of $\psi_{x,g}$, every such row $x \in X_{\text{hon}}$ is labeled by an attribute a_x on which $\mathcal{A}$ made a key-generation query with global identifier g , and so the corrupt attributes together with A_g satisfy P^* , contradicting the admissibility of $\mathcal{A}$.

Otherwise, $\psi_g^\top \mathbf{M}^* \in V_{\text{cor}}$ for every $g \in G$ (i.e., there exists a vector $\mathbf{v}_g = (v_{1,g}, \dots, v_{\ell,g})$ with $v_{x,g} = 0$ for every $x \in X_{\text{hon}}$ such that $(\psi_g - \mathbf{v}_g)^\top \mathbf{M}^* = 0^\top$). The nonzero entries of $\psi_g - \mathbf{v}_g$ are all in rows labeled by attributes in $C \cup A_g$, where C denotes the set of corrupt attributes, and $C \cup A_g$ does not satisfy P^* by the admissibility of $\mathcal{A}$. Because the challenge policy P^* is linearly independent for unauthorized sets, the rows of $\mathbf{M}^*$ labeled by attributes in $C \cup A_g$ are linearly independent, and so $\psi_g - \mathbf{v}_g = \mathbf{0}$. As $\psi_{x,g} = 0$ for every $x \in X_{\text{cor}}$ and $v_{x,g} = 0$ for every $x \in X_{\text{hon}}$, it follows that $\psi_g = \mathbf{0}$ for every $g \in G$. Finally, because we consider unique-attribute-identity security, for each $x \in X_{\text{hon}}$ and $g \in G$ there is at most one key ξ with $a_\xi = a_x$

and $\text{gid}_\xi = g$, so $\psi_{x,g} = \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z)$ for this key (if it exists). Thus, it must hold that $\text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z) = 0$ for every $(x, \xi) \in X_{\text{hon}} \times [q_{\text{KG}}]$ with $a_\xi = a_x$, which contradicts the assumption that D is non-empty.

Wrapping up, we have shown that if $\text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \neq 0$ for some $(x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}]$, $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) = 0$, and $|G| < K$, then there is no condition under which $\tilde{h}$ is identically zero. Thus,

$$\Pr[\text{bad}_{\text{I-b}}] = 0.$$

- **Case II: $\hat{h}_Z$ combines keys for K or more distinct global identifiers:** We now analyze the bad event in which $|G| \geq K$. As argued for Case I-b, including $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms in $\hat{h}_Z$ induces $\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$, which must be cancelled by also including $\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}$ terms in $\hat{h}_Z$ (as these are the only other terms which induce terms of the form $\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi}$). At a high level, we will show that including $\hat{c}_{3,x}^* \hat{\delta}_\xi$ terms for K distinct global identifiers in $\hat{h}_Z$ will, for some row $x \in X_{\text{hon}}$, induce $\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$ for a sufficient number of $i \in [L]$ such that, with high probability, there is some i' induced such that $i' \notin S_x^*$. For i' , the $\hat{c}_{5,x,i'}^* \hat{\beta}_{i',\xi}$ term needed in $\hat{h}_Z$ induces a $\tilde{r}'_{x,i'} \tilde{\beta}_{i',\xi}$ term as opposed to a $\tilde{r}_x \tilde{y}_{i',\xi} \tilde{\beta}_{i',\xi}$ term (because $i' \notin S_x^*$), and thus cancellation cannot occur and $\tilde{h}$ is not identically zero.

Because the $\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi}$ terms we analyze are identical to those used in Case I-b, we directly proceed with the cancellation argument. Let $C = \{(x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}] : \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \neq 0\}$ be the set of indices for $\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}_Z$ and define the following sets:

$$\begin{aligned} C' &= \{(x, i, \xi) : (x, i, j, \xi) \in C \wedge i = j\}, & C'' &= \{(x, i, j, \xi) \in C : i \neq j\}, \\ C_{S^*}' &= \{(x, i, \xi) \in C' : i \in S_x^*\}, & C_{S^*}'' &= \{(x, i, j, \xi) \in C'' : i \in S_x^*\}. \end{aligned}$$

As in Case I-b, the challenger's formal polynomial can be written as

$$\begin{aligned} \tilde{h} &= \tilde{\mu} + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}^* \hat{\delta}_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \left(\tilde{\alpha}_{\xi} + \tilde{y}_{\xi} \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi} \right) \\ &+ \sum_{(x,i,\xi) \in C_{S^*}'} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi} \right) + \sum_{(x,i,\xi) \in C' \setminus C_{S^*}'} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}'_{x,i} \tilde{\beta}_{i,\xi} \right) \\ &+ \sum_{(x,i,j,\xi) \in C_{S^*}''} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{j,\xi} \right) + \sum_{(x,i,j,\xi) \in C'' \setminus C_{S^*}''} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi} \right) \end{aligned} \quad (4.14)$$

where $\tilde{\mu}$ is a polynomial over $\mathcal{V}$ such that $\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = \text{coeff}_{\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = 0$ for all $(x, i, j, \xi) \in X_{\text{hon}} \times [L]^2 \times [q_{\text{KG}}]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}$ terms). In particular, because $\langle m_b^* \rangle$ must have been created before the challenge phase, neither $\tilde{r}_x$ nor any $\tilde{r}'_{x,i}$ appear in $\tilde{m}_b^*$ which was absorbed into $\tilde{\mu}$.

By inspection of Eq. (4.14) and using the above fact on $\tilde{\mu}$, we have

$$\begin{aligned} \text{coeff}_{\tilde{r}'_{x,i} \tilde{\beta}_{i,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) & \text{for } (x, i, \xi) \in C' \setminus C_{S^*}', \\ \text{coeff}_{\tilde{r}_x \tilde{y}_{i,\xi} \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) & \text{for } (x, i, j, \xi) \in C_{S^*}'', \\ \text{coeff}_{\tilde{r}'_{x,i} \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{j,\xi}}(\hat{h}_Z) & \text{for } (x, i, j, \xi) \in C'' \setminus C_{S^*}''. \end{aligned}$$

Thus, if C_{S^*}' , $C'' \setminus C_{S^*}'$, or $C' \setminus C_{S^*}'$ are non-empty, then $\tilde{h} \not\equiv 0$.

Now suppose that C''_{S^*} , $C'' \setminus C''_{S^*}$, and $C' \setminus C'_{S^*}$ are empty. We can simplify Eq. (4.14) to

$$\begin{aligned} \tilde{h} = & \tilde{\mu} + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}^* \delta_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \left(\tilde{\alpha}_{\zeta_x} + \tilde{y}_{\zeta_x} \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \right) \\ & + \sum_{(x,i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \left(\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \right). \end{aligned} \quad (4.15)$$

Rearranging Eq. (4.15) further, we have that

$$\begin{aligned} \tilde{h} = & \tilde{\mu}' + \sum_{(x,i,\xi) \in U \cap C'_{S^*}} \tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_{3,x}^* \delta_\xi}(\hat{h}_Z) + \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \right) \\ & + \sum_{(x,i,\xi) \in U \setminus C'_{S^*}} \text{coeff}_{\hat{c}_{3,x}^* \delta_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi} + \sum_{(x,i,\xi) \in C'_{S^*} \setminus U} \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \cdot \tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi}, \end{aligned} \quad (4.16)$$

where $U = \bigcup_{(x,\xi) \in D} \{(x,i,\xi) : i \in S_\xi\}$ is the set of indices (x,i,ξ) such that $(x,\xi) \in D$ and $i \in S_\xi$ (i.e., the indices that appear in the left-hand sum in Eq. (4.15)), and where

$$\tilde{\mu}' = \tilde{\mu} + \sum_{(x,\xi) \in D} \text{coeff}_{\hat{c}_{3,x}^* \delta_\xi}(\hat{h}_Z) \cdot \tilde{r}_x \left(\tilde{\alpha}_{\zeta_x} + \tilde{y}_{\zeta_x} \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j \right).$$

Notably, we have absorbed the $\tilde{r}_x \tilde{\alpha}_{\zeta_x}$ and $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{u}_j$ terms into $\tilde{\mu}'$ so that $\tilde{\mu}'$ contains all the terms that are not core to the analysis of the $|G| \geq K$ case. By construction, we have that $\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi}}(\tilde{\mu}') = 0$ for all $(x,i,j,\xi) \in X_{\text{hon}} \times [L]^2 \times [q_{KG}]$.

Intuitively, we can see from Eq. (4.16) that the only cancellations can occur in $U \cap C'_{S^*}$. That is, the sets $U \setminus C'_{S^*}$ and $C'_{S^*} \setminus U$ must be empty (which we will formally argue below) and thus $C'_{S^*} = U$ for the bad event to occur. We conclude by arguing that the adversary has a negligible probability of producing an $\hat{h}_Z$ such that $C'_{S^*} = U$.

Proceeding with the argument outlined above, if $U \neq C'_{S^*}$ then there exists some $(x,i,\xi) \in (U \setminus C'_{S^*}) \cup (C'_{S^*} \setminus U)$. It follows that $\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi}}(\tilde{h}) = \text{coeff}_{\hat{c}_{3,x}^* \delta_\xi}(\hat{h}_Z) \neq 0$ or $\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{\beta}_{i,\xi}}(\tilde{h}) = \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\beta}_{i,\xi}}(\hat{h}_Z) \neq 0$, so $\tilde{h} \neq 0$ in this case.

It remains to bound the probability that $C'_{S^*} = U$, which itself is bounded by the probability that $U \subseteq C'_{S^*}$. First we recall that $C'_{S^*} = \{(x,i,\xi) \in C' : i \in S_x^*\}$ and $U = \bigcup_{(x,\xi) \in D} \{(x,i,\xi) : i \in S_\xi\}$, and, for each row $x \in X_{\text{hon}}$, we let $D_x = \{\xi \in [q_{KG}] : (x,\xi) \in D\}$ denote the set of keys ξ issued on the attribute a_x such that $\hat{h}_Z$ has a nonzero $\hat{c}_{3,x}^* \delta_\xi$ term. Because $|D| \geq |G| \geq K$ and $\ell \leq M$, there exists some row $x \in X_{\text{hon}}$ with $|D_x| \geq K/M = K^*$, and every key $\xi \in D_x$ is issued on the same attribute a_x .

Per the change added in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$, the sets S_x^* for $x \in X_{\chi-1,\text{hon}}$ are sampled after the $(\chi-1)^{\text{th}}$ post-challenge zero-test query, and so no zero-test query is answered between their sampling and the current query. Furthermore, $X_{\text{hon}} = X_{\chi,\text{hon}} \subseteq X_{\chi-1,\text{hon}}$, as corrupt authorities remain corrupt and, by admissibility, no authority-setup query is made on an attribute supplied during the challenge phase. Because S^* influences $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$, the adversary has not received any information on S_x^* for $x \in X_{\text{hon}}$, so for any row $x \in X_{\text{hon}}$ with $|D_x| \geq K^*$ (e.g., the smallest such row) we have that

$$\Pr[\text{bad}_{\Pi}] \leq \Pr[U \subseteq C'_{S^*}] \leq \Pr\left[\bigcup_{\xi \in D_x} S_\xi \subseteq S_x^*\right] \leq \left(\frac{L_{\text{ct}}}{L}\right)^{\frac{1}{K^*}},$$

where the final inequality follows from the halting condition added in hybrid $\text{Hyb}_1^{(b)}$ which asks that there is no collection of sets $S_{\xi_1}, \dots, S_{\xi_{K^*}}$ sampled during key-generation queries on the same attribute with union size less than or equal to L/κ . Since $|D_x| \geq K^*$ and every key in D_x is issued on a_x , this condition applies to any K^* elements of D_x and yields $|\bigcup_{\xi \in D_x} S_{\xi}| > L/\kappa$.

Finally, recall that $L - L_{\text{ct}} = \omega(\log \lambda)$ and $\kappa \geq 2$ is a constant. Consider any $c_1 > 0$. Because $L - L_{\text{ct}} = \omega(\log \lambda)$, there exists some $d_1 \in \mathbb{N}$ such that $L - L_{\text{ct}} \geq c_1 \log \lambda$ for all $\lambda \geq d_1$. Let $c = c_1 \log e / \kappa$. We have that,

$$\begin{aligned} \Pr[\text{bad}_{\Pi}] &\leq \left(1 - \frac{L - L_{\text{ct}}}{L}\right)^{\frac{L}{\kappa}} \\ &\leq \exp\left(-\frac{L - L_{\text{ct}}}{\kappa}\right) \\ &\leq \exp\left(-\frac{c_1 \log \lambda}{\kappa}\right) && \text{for } \lambda \geq d_1 \\ &= \lambda^{-c_1 \log e / \kappa} && \text{for } \lambda \geq d_1 \\ &= \lambda^{-c} && \text{for } \lambda \geq d_1. \end{aligned}$$

Note that since $c_1 > 0$ was arbitrary, $c = c_1 \log e / \kappa$ can take any positive value, and so there exists a negligible function $\text{negl}(\cdot)$ such that

$$\Pr[\text{bad}_{\Pi}] = \text{negl}(\lambda).$$

- **Case III: $\text{coeff}_{\hat{c}_{5,x,i}^* \hat{\nu}}(\hat{h}_Z) \neq 0$ for some $(x, i, \hat{\nu}) \in X_{\text{hon}} \times [L] \times (\{1\} \cup \hat{\mathcal{V}}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]})$:** This final case considers the remaining challenge-set-dependent terms not covered in the previous cases.

Explicitly, this case occurs if, for $(x, i) \in X_{\text{hon}} \times [L]$, $\text{coeff}_{\hat{c}_{5,x,i}^*}(\hat{h}_Z) \neq 0$, $\text{coeff}_{\hat{c}_{5,x,i}^* \hat{u}_j}(\hat{h}_Z) \neq 0$ for some $j \in [K]$, or $\text{coeff}_{\hat{c}_{5,x,i}^* \delta_{\xi}}(\hat{h}_Z) \neq 0$ for some $\xi \in [q_{\text{KG}}]$. We consider these as sub-cases and analyze them below:

- **Case III-a $\text{coeff}_{\hat{c}_{5,x,i}^*}(\hat{h}_Z) \neq 0$ for some $(x, i) \in X_{\text{hon}} \times [L]$:** A $\hat{c}_{5,x,i}^*$ term in $\hat{h}_Z$ induces an $\tilde{r}_x \tilde{y}_{i,\zeta_x}$ or $\tilde{r}'_{x,i}$ term in $\tilde{h}$. We argue that no other terms induce $\tilde{r}_x \tilde{y}_{i,\zeta_x}$ or $\tilde{r}'_{x,i}$ terms in $\tilde{h}$, and thus no cancellation occurs in this case. From Eq. (4.3), we have that no other term in $\hat{h}_Z$ induces a term of the form $\tilde{r}_x \tilde{y}_{i,\zeta_x}$ and thus the only remaining option to induce it is to pair an element $\hat{v}_1 \in \hat{\mathcal{V}}_1$ and $\hat{v}_2 \in \hat{\mathcal{V}}_2$. If $\hat{v}_1$ is $\hat{c}_{3,x}^*$, then this induces an $\tilde{r}_x$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains a (solo) $\tilde{y}_{i,\zeta_x}$ term. If $\hat{v}_1$ is $\hat{y}_{i,\zeta_x}$, then this induces a $\tilde{y}_{i,\zeta_x}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains an $\tilde{r}_x$ term. The same holds if $\hat{v}_1$ is $\hat{y}_{i',\zeta'}$ for some $(i', \zeta') \neq (i, \zeta_x)$, which induces a $\tilde{y}_{i',\zeta'}$ term or a constant, or $\hat{c}_{5,x',i'}^*$ for some $(x', i') \neq (x, i)$, which induces an $\tilde{r}_{x'} \tilde{y}_{i',a_{x'}}$ or $\tilde{r}'_{x',i'}$ term that, as $(x', i') \neq (x, i)$, is neither $\tilde{r}_x \tilde{y}_{i,\zeta_x}$ nor $\tilde{r}'_{x,i}$. If $\hat{v}_1$ is $\hat{y}_{\zeta'}$ for some $\zeta' \in [q_{\text{AS}}]$, $\hat{c}_{4,x'}^*$ for some $x' \in [\ell]$, or $\hat{c}_{3,x'}^*$ for some $x' \neq x$, then each term this induces is a constant or contains a $\tilde{y}_{\zeta'}$, $\tilde{w}_k$, or $\tilde{r}_{x'}$ variable for $x' \neq x$, which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$, none of which contains an $\tilde{r}_x$ term) cannot equal $\tilde{r}_x \tilde{y}_{i,\zeta_x}$. Finally, we have that no other term in $\hat{h}_Z$ induces a (solo) $\tilde{r}'_{x,i}$. In particular, a $\hat{c}_1^*$ term induces neither an $\tilde{r}_x \tilde{y}_{i,\zeta_x}$ nor an $\tilde{r}'_{x,i}$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{r}_x$ nor any $\tilde{r}'_{x,i}$ appears in a $\tilde{\mathcal{T}}$ entry before then.
- **Case III-b $\text{coeff}_{\hat{c}_{5,x,i}^* \hat{u}_j}(\hat{h}_Z) \neq 0$ for some $(x, i, j) \in X_{\text{hon}} \times [L] \times [K]$:** A $\hat{c}_{5,x,i}^* \hat{u}_j$ term in $\hat{h}_Z$ induces an $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{u}_j$ or $\tilde{r}'_{x,i} \tilde{u}_j$ term in $\tilde{h}$. We argue that no other terms induce these terms in $\tilde{h}$. From Eq. (4.3), we can see that the only remaining option to induce an $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{u}_j$ or $\tilde{r}'_{x,i} \tilde{u}_j$ term, other than by pairing $\hat{c}_{5,x,i}^*$ with $\hat{u}_j$ itself, is to pair some other element $\hat{v}_1 \in \hat{\mathcal{V}}_1$ with $\hat{v}_2 \in \hat{\mathcal{V}}_2$. If $\hat{v}_1$ is $\hat{c}_{3,x}^*$, then this induces an $\tilde{r}_x$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains a (solo) $\tilde{y}_{i,\zeta_x} \tilde{u}_j$ term. If $\hat{v}_1$ is $\hat{y}_{i,\zeta_x}$, then this induces a $\tilde{y}_{i,\zeta_x}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains an $\tilde{r}_x \tilde{u}_j$ term, and the same holds for any other $\hat{y}_{i',\zeta'}$, which induces a $\tilde{y}_{i',\zeta'}$ term or a constant. If $\hat{v}_1$ is $\hat{c}_{5,x',i'}^*$ for some $(x', i') \neq (x, i)$, then this induces an $\tilde{r}_{x'} \tilde{y}_{i',a_{x'}}$ or $\tilde{r}'_{x',i'}$ term in $\tilde{h}$,

which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$) cannot equal $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{u}_j$ or $\tilde{r}'_{x,i} \tilde{u}_j$ since $(x', i') \neq (x, i)$. If $\hat{v}_1$ is $\hat{y}_{\zeta'}$ for some $\zeta' \in [q_{AS}]$, $\hat{c}_{4,x'}^*$ for some $x' \in [L]$, or $\hat{c}_{3,x'}^*$ for some $x' \neq x$, then each term this induces is a constant or contains a $\tilde{y}_{\zeta'}$, $\tilde{w}_k$, or $\tilde{r}_{x'}$ variable for $x' \neq x$, which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$) cannot equal $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{u}_j$ or $\tilde{r}'_{x,i} \tilde{u}_j$. Finally, we have that $\hat{u}_j$ is the only element of $\hat{\mathcal{V}}_2$ which induces a (solo) $\tilde{u}_j$ term (as $\hat{h}_Z$ contains no $\hat{\delta}_\xi$ for keys issued on corrupt attributes), so no other pairing can induce the required $\tilde{u}_j$ factor either. In particular, like Case III-a, a $\hat{c}_1^*$ term induces neither an $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{u}_j$ nor an $\tilde{r}'_{x,i} \tilde{u}_j$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{r}_x$ nor any $\tilde{r}'_{x,i}$ appears in a $\tilde{\mathcal{T}}$ entry before then.

- **Case III-c coeff $\hat{c}_{5,x,i}^* \hat{\delta}_\xi (\hat{h}_Z) \neq 0$ for some $(x, i, \xi) \in X_{\text{hon}} \times [L] \times [q_{KG}]$:** For any $j \in S_\xi$, a $\hat{c}_{5,x,i}^* \hat{\delta}_\xi$ term in $\hat{h}_Z$ induces an $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ term in $\tilde{h}$, where ζ is the authority on which the key ξ was issued (which is honest, as $\hat{h}_Z$ contains no $\hat{\delta}_\xi$ for keys issued on corrupt attributes). We argue that no other terms induce these terms in $\tilde{h}$. From Eq. (4.3), we can see that the only remaining option to induce an $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ term, other than by pairing $\hat{c}_{5,x,i}^*$ with $\hat{\delta}_\xi$ itself, is to pair some other element $\hat{v}_1 \in \hat{\mathcal{V}}_1$ with $\hat{v}_2 \in \hat{\mathcal{V}}_2$. If $\hat{v}_1$ is $\hat{c}_{3,x}^*$, then this induces an $\tilde{r}_x$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains a (solo) $\tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ term. If $\hat{v}_1$ is $\hat{y}_{i,\zeta_x}$ or $\hat{y}_{j,\zeta}$, then this induces a $\tilde{y}_{i,\zeta_x}$ or $\tilde{y}_{j,\zeta}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains an $\tilde{r}_x$ or $\tilde{r}'_{x,i}$ term, and the same holds for any other $\hat{y}_{i',\zeta'}$, which induces a $\tilde{y}_{i',\zeta'}$ term or a constant. If $\hat{v}_1$ is $\hat{c}_{5,x',i'}^*$ for some $(x', i') \neq (x, i)$, then this induces an $\tilde{r}_{x'} \tilde{y}_{i',\zeta'} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x',i'} \tilde{\beta}_{j,\xi}$ term in $\tilde{h}$, which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$) cannot equal $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ since $(x', i') \neq (x, i)$, except when $\zeta = \zeta_x$ and $(x', i') = (x, j)$ with $j \neq i$, in which case $\hat{c}_{5,x,j}^*$ induces $\tilde{r}_x \tilde{y}_{j,\zeta_x}$ and $\hat{v}_2$ would instead have to induce $\tilde{y}_{i,\zeta_x} \tilde{\beta}_{j,\xi}$ with $i \neq j$, which no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ does, as $\hat{\delta}_\xi$ only induces $\tilde{y}_{i',\zeta} \tilde{\beta}_{i',\xi}$ terms with matched indices. If $\hat{v}_1$ is $\hat{y}_{\zeta'}$ for some $\zeta' \in [q_{AS}]$, $\hat{c}_{4,x'}^*$ for some $x' \in [L]$, or $\hat{c}_{3,x'}^*$ for some $x' \neq x$, then each term this induces is a constant or contains a $\tilde{y}_{\zeta'}$, $\tilde{w}_k$, or $\tilde{r}_{x'}$ variable for $x' \neq x$, which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$) cannot equal $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ or $\tilde{r}'_{x,i} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$. Finally, as argued in Case I-b, $\hat{\beta}_{j,\xi}$ and $\hat{\delta}_\xi$ are the only elements of $\hat{\mathcal{V}}_2$ which induce a $\tilde{\beta}_{j,\xi}$ term, and the former would have to be paired with an element of $\hat{\mathcal{V}}_1$ inducing $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta}$ or $\tilde{r}'_{x,i} \tilde{y}_{j,\zeta}$, of which there are none, so no other pairing can induce the required $\tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ factor either. We note in particular that $\tilde{\beta}_{j,\xi}$ is unique to the key ξ , so these terms cannot be cancelled by $\hat{c}_{5,x,i}^* \hat{\delta}_{\xi'}$ terms for any $\xi' \neq \xi$. Note that, like Case III-a, a $\hat{c}_1^*$ term induces neither an $\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ nor an $\tilde{r}'_{x,i} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{r}_x$ nor any $\tilde{r}'_{x,i}$ appears in a $\tilde{\mathcal{T}}$ entry before then.

Summarizing the analysis above, we have

$$\begin{aligned}
\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\zeta_x}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^*}(\hat{h}_Z) && \text{for } x \in X_{\text{hon}}, i \in S_x^*, \\
\text{coeff}_{\tilde{r}'_{x,i}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^*}(\hat{h}_Z) && \text{for } x \in X_{\text{hon}}, i \in [L] \setminus S_x^*, \\
\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{u}_j}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^* \hat{u}_j}(\hat{h}_Z) && \text{for } x \in X_{\text{hon}}, i \in S_x^*, j \in [K], \\
\text{coeff}_{\tilde{r}'_{x,i} \tilde{u}_j}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^* \hat{u}_j}(\hat{h}_Z) && \text{for } x \in X_{\text{hon}}, i \in [L] \setminus S_x^*, j \in [K], \\
\text{coeff}_{\tilde{r}_x \tilde{y}_{i,\zeta_x} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\delta}_\xi}(\hat{h}_Z) && \text{for } x \in X_{\text{hon}}, i \in S_x^*, \xi \in [q_{KG}], j \in S_\xi, \\
\text{coeff}_{\tilde{r}'_{x,i} \tilde{y}_{j,\zeta} \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{5,x,i}^* \hat{\delta}_\xi}(\hat{h}_Z) && \text{for } x \in X_{\text{hon}}, i \in [L] \setminus S_x^*, \xi \in [q_{KG}], j \in S_\xi,
\end{aligned}$$

where, in the last two lines, ξ ranges over keys issued on honest attributes and ζ is the authority on which ξ was issued. It follows that if any of the Case III terms are included in $\hat{h}_Z$ for a row $x \in X_{\text{hon}}$, then $\tilde{h}$ has a corresponding term and thus $\tilde{h}$ is never identically zero. Thus,

$$\Pr[\text{bad}_{\text{III}}] = 0.$$

Combining the case analysis above, we overall have that

$$\begin{aligned}\Pr[\text{bad}] &\leq \Pr[\text{bad}_{\text{I-a}}] + \Pr[\text{bad}_{\text{I-b}}] + \Pr[\text{bad}_{\text{II}}] + \Pr[\text{bad}_{\text{III}}] \\ &= \text{negl}(\lambda).\end{aligned}$$

□

Lemma 4.17. *For all $\lambda \in \mathbb{N}$ and all $\chi \in [q_{\text{ZT}}]$, $\Pr[\text{Hyb}_{5,\chi}^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,\chi,\text{int}}^{(b)}(\lambda)]$.*

Proof. All that differs between the two hybrids is when the challenger samples the sets S_x^* for $x \in X_{\chi-1,\text{hon}}$ and performs the substitution (via the ring homomorphism $\text{SubS}_{X,Z}$ for the appropriate $X \subseteq X_{\chi-1,\text{hon}}$ and set Z of corrupt authorities) on the entries of $\tilde{\mathcal{T}}$. Specifically, in hybrid $\text{Hyb}_{5,\chi}^{(b)}$ this step is performed for all $x \in X_{\chi-1,\text{hon}}$ immediately after the $(\chi-1)^{\text{th}}$ post-challenge zero-test query (or at the start of the post-challenge query phase if $\chi = 1$), while in hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$ it is instead performed for $x \in X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}$ immediately before the χ^{th} post-challenge zero-test query and for $x \in X_{\chi,\text{hon}}$ immediately after it. We will argue that $\mathcal{A}$'s view is identical in both hybrids.

We first note that, starting in hybrid $\text{Hyb}_4^{(b)}$ (and for all remaining hybrids), the zero-test oracle answers a query on a handle $\langle h \rangle$ by testing whether $\tilde{h} \equiv 0$ (or alternatively not checking $\tilde{h}$ at all), and nothing derived from exponents from the table $\mathcal{T}$ is given to $\mathcal{A}$ other than the authority secret keys $\text{msk}_\zeta = (\alpha_\zeta, y_\zeta, \{y_{i,\zeta}\}_{i \in [L]})$ of *corrupt* authorities ζ , which are sampled independently of the sets S_x^* for $x \in X_{\chi-1,\text{hon}}$. It follows that the sets S_x^* for $x \in X_{\chi-1,\text{hon}}$ influence $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$.

Now, observe that $\mathcal{A}$'s view through the $(\chi-1)^{\text{th}}$ post-challenge zero-test query (or through the end of the challenge phase if $\chi = 1$) is identical in both hybrids, as the two hybrids perform identical operations up to that point. We refer to the point immediately following that query (or the start of the post-challenge query phase if $\chi = 1$) as the *substitution point*, at which hybrid $\text{Hyb}_{5,\chi}^{(b)}$ samples and substitutes for $(X_{\chi-1,\text{hon}}, Z_{\chi-1})$ while hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$ does not. In particular, the answers to the first $\chi-1$ post-challenge zero-test queries are identical in both hybrids, and the handles present in $\tilde{\mathcal{T}}$ at the substitution point are the same in both hybrids. We also recall that $X_{\chi,\text{hon}} \subseteq X_{\chi-1,\text{hon}}$, so that $X_{\chi-1,\text{hon}}$ is the disjoint union of $X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}$ and $X_{\chi,\text{hon}}$.

It remains to argue that the χ^{th} post-challenge zero-test query is answered identically and that the entries of $\tilde{\mathcal{T}}$ coincide in both hybrids from that query onwards. Consider any handle $\langle h \rangle$ present in $\tilde{\mathcal{T}}$ and let $\tilde{h}'$ denote the entry it would have *at the same point in time* in the absence of any substitution for the rows in $X_{\chi-1,\text{hon}}$ (i.e., the entry obtained by recording $\langle c_{5,x,i}^* \rangle \mapsto \tilde{r}'_{x,i}$ for every $x \in X_{\chi-1,\text{hon}}$ and $i \in [L]$), which is precisely its entry in hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$ up until the substitution for $X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}$. Note that we have the requirement on time because $\tilde{h}'$ can be rewritten by the authority-corruption queries. We write $\text{SubS}_Z = \text{SubS}_{X_{\chi-1,\text{hon}},Z}$ (because hybrid $\text{Hyb}_{5,\chi}^{(b)}$ samples and substitutes with $X_{\chi-1,\text{hon}}$) and let t_0 denote the substitution point. By time independence (Claim 4.14, applied with $\chi-1$ in place of χ), at every time $t \geq t_0$ and for every $Z \supseteq Z_t$, SubS_Z maps $\tilde{r}'_{x,i} \mapsto \text{Corr}_Z(\tilde{r}_x \tilde{y}_{i,\zeta_x})$ for $x \in X_{\chi-1,\text{hon}}$ and $i \in S_x^*$, and every other variable to itself. We now argue, by induction on the steps that create or modify entries of $\tilde{\mathcal{T}}$ from the substitution point up to the χ^{th} post-challenge zero-test query, that $\tilde{\mathcal{T}}(\langle h \rangle) = \text{SubS}_{Z_t}(\tilde{h}')$ holds at every time $t \geq t_0$, for every handle $\langle h \rangle$ in $\tilde{\mathcal{T}}$, in hybrid $\text{Hyb}_{5,\chi}^{(b)}$ (recall that the corresponding entry in hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$ is $\tilde{h}'$ itself throughout this period, by the definition of $\tilde{h}'$).

For the base case, consider a handle $\langle h \rangle$ that is stored in $\tilde{\mathcal{T}}$ at the substitution point, where $Z_t = Z_{\chi-1}$. The base case holds by definition, as hybrid $\text{Hyb}_{5,\chi}^{(b)}$ samples and substitutes for $(X_{\chi-1,\text{hon}}, Z_{\chi-1})$ there, which replaces $\tilde{h}'$ in $\tilde{\mathcal{T}}$ with $\text{SubS}_{Z_{\chi-1}}(\tilde{h}')$. For the inductive step, consider a creation or modification of $\tilde{\mathcal{T}}$ performed after the substitution point and before the χ^{th} post-challenge zero-test query, and let t^- and t^+ denote the times immediately before and after it. Now suppose that $\tilde{\mathcal{T}}(\langle h \rangle) = \text{SubS}_{Z_{t^-}}(\tilde{h}')$ for all handles $\langle h \rangle$ in the table at time t^- . We go over each possible change to $\tilde{\mathcal{T}}$ below:

- **Post-challenge authority-setup queries:** On the ζ^{th} authority-setup query, the challenger adds the entries $\langle \alpha_\zeta \rangle \mapsto \tilde{\alpha}_\zeta$, $\langle y_\zeta \rangle \mapsto \tilde{y}_\zeta$, and $\langle y_{i,\zeta} \rangle \mapsto \tilde{y}_{i,\zeta}$ for each $i \in [L]$ to $\tilde{\mathcal{T}}$. The corrupted authority set is unchanged, so $Z_{t^+} = Z_{t^-}$, and no entry contains any of the variables $\{\tilde{r}'_{x,i}\}_{(x,i) \in X_{\chi-1,\text{hon}} \times [L]}$, and so, by invariance (Claim 4.14), all added entries are unaffected by the $\text{SubS}_{Z_{t^+}}$ ring homomorphism.

- **Post-challenge authority-corruption queries:** A corruption query replaces every entry $\langle h \rangle \mapsto \tilde{h}$ of $\tilde{\mathcal{T}}$ with $\langle h \rangle \mapsto \text{Corr}_{Z_{t^+}}(\tilde{h})$ for the updated set Z_{t^+} of corrupted authorities. Let $\tilde{h}'$ denote the entry $\langle h \rangle$ would have at time t^- in the absence of any substitution, so that the corresponding entry at time t^+ is $\text{Corr}_{Z_{t^+}}(\tilde{h}')$. As $Z_{t^-} \subseteq Z_{t^+}$, the induction hypothesis and corruption commutativity (Claim 4.14) give us

$$\tilde{\mathcal{T}}(\langle h \rangle) = \text{Corr}_{Z_{t^+}}(\text{SubS}_{Z_{t^-}}(\tilde{h}')) = \text{SubS}_{Z_{t^+}}(\text{Corr}_{Z_{t^+}}(\tilde{h}')).$$

- **Post-challenge key-generation queries:** On the ξ^{th} key-generation query, the challenger adds the entries $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ for each $i \in S_\xi$ and $\langle \delta_\xi \rangle \mapsto \text{Corr}_{Z_{t^+}}(\tilde{\alpha}_\xi + \tilde{y}_\xi \sum_{j \in [K]} \text{gid}_\xi^{j-1} \tilde{u}_j + \sum_{i \in S_\xi} \tilde{y}_{i,\xi} \tilde{\beta}_{i,\xi})$ to $\tilde{\mathcal{T}}$. Again $Z_{t^+} = Z_{t^-}$ and neither entry contains any of the variables $\{\tilde{r}'_{x,i}\}_{(x,i) \in X_{\chi-1,\text{hon}} \times [L]}$, so, by invariance (Claim 4.14), both are unaffected by the $\text{SubS}_{Z_{t^+}}$ ring homomorphism.
- **Post-challenge generic group oracle queries:** All three of these queries also leave the corrupt set unchanged, so $Z_{t^+} = Z_{t^-}$. The analysis of the generic group oracle queries is as follows:

- **BGGM.Encode(x, v):** The challenger adds the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, which is again unaffected by the $\text{SubS}_{Z_{t^+}}$ ring homomorphism.
- **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** Recall that the challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that $\text{SubS}_{Z_{t^+}}$ is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1) + \text{SubS}_{Z_{t^+}}(\tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1 + \tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}').$$

- **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** The challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that $\text{SubS}_{Z_{t^+}}$ is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1) \cdot \text{SubS}_{Z_{t^+}}(\tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}'_1 \cdot \tilde{h}'_2) = \text{SubS}_{Z_{t^+}}(\tilde{h}').$$

In each case, we have that $\tilde{\mathcal{T}}(\langle h \rangle) = \text{SubS}_{Z_{t^+}}(\tilde{h}')$ at time t^+ in hybrid $\text{Hyb}_{5,\chi}^{(b)}$.

Next, consider the χ^{th} post-challenge zero-test query on a handle $\langle h \rangle$ with associated adversary formal polynomial $\hat{h}_Z$ (as defined in hybrid $\text{Hyb}_{5,\chi}^{(b)}$). Immediately before this query, hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$ samples and substitutes for $(X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}, Z_\chi)$, so the entry of $\langle h \rangle$ in hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$ is $\tilde{h} = \text{SubS}_{X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}, Z_\chi}(\tilde{h}')$, while, by the induction above and parallel composition (Claim 4.14), its entry in hybrid $\text{Hyb}_{5,\chi}^{(b)}$ is $\text{SubS}_{Z_\chi}(\tilde{h}') = \text{SubS}_{X_{\chi,\text{hon}}, Z_\chi}(\text{SubS}_{X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}, Z_\chi}(\tilde{h}')) = \text{SubS}_{X_{\chi,\text{hon}}, Z_\chi}(\tilde{h})$. As the behavior added in hybrid $\text{Hyb}_{5,\chi}^{(b)}$ applies to this query, if $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \neq 0$ or $\text{coeff}_{\hat{c}_{5,x,i}^*}(\hat{h}_Z) \neq 0$ for some $(x, i, \hat{v}) \in X_{\chi,\text{hon}} \times [L] \times (\{1\} \cup \hat{\mathcal{V}}_2)$, then both hybrids return “non-zero” without accessing the table $\tilde{\mathcal{T}}$ at all, and the query is answered identically. Otherwise, $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) = \text{coeff}_{\hat{c}_{5,x,i}^*}(\hat{h}_Z) = 0$ for all $(x, i, \hat{v}) \in X_{\chi,\text{hon}} \times [L] \times (\{1\} \cup \hat{\mathcal{V}}_2)$. In particular, by inspection of Eq. (4.3) (where, in hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$, $\hat{c}_{5,x,i}^* \mapsto \tilde{r}'_{x,i}$ for every $(x, i) \in X_{\chi,\text{hon}} \times [L]$ at the time of this query), $\hat{c}_{5,x,i}^*$ is the only variable in $\hat{\mathcal{V}}$ that can induce a $\tilde{r}'_{x,i}$ variable in $\tilde{h}$, and thus $\tilde{h}$ does not contain any term involving any of the variables $\{\tilde{r}'_{x,i}\}_{(x,i) \in X_{\chi,\text{hon}} \times [L]}$. Here, we recall from Lemma 4.16 that applying Expd_{Z_χ} to $\tilde{\mathcal{T}}(\langle h \rangle)$ does not change the associated challenger formal polynomial (i.e., $\tilde{\mathcal{T}}(\langle h \rangle)$ and $\hat{h}_Z$ have the same image under Eq. (4.3)). Hence, by invariance (Claim 4.14), $\text{SubS}_{X_{\chi,\text{hon}}, Z_\chi}(\tilde{h}) = \tilde{h}$, and so the χ^{th} query is again answered identically whether or not the substitution for $X_{\chi,\text{hon}}$ has already been applied.

It remains to argue that the entries of $\tilde{\mathcal{T}}$ coincide in both hybrids from the point immediately following the χ^{th} query onwards, from which it follows that the remaining post-challenge zero-test queries are also answered identically. Immediately after the χ^{th} query, hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$ samples and substitutes for $(X_{\chi,\text{hon}}, Z_\chi)$. As no handle is created or modified by the χ^{th} query itself, this replaces each entry $\tilde{h} = \text{SubS}_{X_{\chi-1,\text{hon}} \setminus X_{\chi,\text{hon}}, Z_\chi}(\tilde{h}')$ with $\text{SubS}_{X_{\chi,\text{hon}}, Z_\chi}(\tilde{h}) = \text{SubS}_{Z_\chi}(\tilde{h}')$ by parallel composition (Claim 4.14), which is exactly the entry of $\langle h \rangle$ in hybrid $\text{Hyb}_{5,\chi}^{(b)}$. From this point onwards, both

hybrids perform identical operations starting from identical tables, and thus $\mathcal{A}$'s view is identical in the two hybrids. It follows that

$$\Pr[\text{Hyb}_{5,\chi}^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,\chi,\text{int}}^{(b)}(\lambda)]. \quad \square$$

Lemma 4.18. *For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_{5,q_{\text{ZT}},\text{int}}^{(0)}(\lambda)] = \Pr[\text{Hyb}_{5,q_{\text{ZT}},\text{int}}^{(1)}(\lambda)]$.*

Proof. The lemma follows from the fact that both hybrids operate independently of the challenge bit b . In particular, any zero-test query dependent on the challenge-message-dependent component (which is the only component dependent on b) must have that $\text{coeff}_{\hat{c}_1^*}(\hat{h}_Z) \neq 0$, and the oracle only ever returns “non-zero” in this case (as the forced-“non-zero” rule applies to every post-challenge zero-test query, and no pre-challenge query can involve $\hat{c}_1^*$). $\square$

The theorem follows by applying a standard hybrid argument over [Lemmas 4.10 to 4.13](#) and [4.15 to 4.18](#). In particular,

$$\begin{aligned} |\Pr[b^* = 1 : b = 0] - \Pr[b^* = 1 : b = 1]| &= |\Pr[\text{Hyb}_0^{(0)}(\lambda)] - \Pr[\text{Hyb}_0^{(1)}(\lambda)]| \\ &\leq 2(3 + q_{\text{ZT}}) \cdot \text{negl}(\lambda) \\ &= \text{negl}(\lambda), \end{aligned}$$

where we have used the fact that $q_{\text{ZT}} = \text{poly}(\lambda)$. $\square$

Parameter instantiation. We now give one example of how to set the parameters of [Construction 4.7](#) so that [Theorem 4.8](#) yields inverse-polynomial error and the conditions of [Theorem 4.9](#) are satisfied. We stress that there is nothing special about our parameter selection; any choice of L , L_{sk} , L_{ct} , and K satisfying the conditions suffices. Take

$$L = M\lambda \quad , \quad K = M^2\lambda \quad , \quad L_{\text{sk}} = \lambda^{\frac{1}{3}} \quad , \quad L_{\text{ct}} = M\lambda - \lambda^{\frac{1}{3}}.$$

Per [Theorem 4.8](#), we have that [Construction 4.7](#) is statistically correct with error

$$\varepsilon(\lambda) = M \cdot L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L}\right) = M \cdot \lambda^{\frac{1}{3}} \cdot \frac{\lambda^{\frac{1}{3}}}{M\lambda} = \lambda^{-\frac{1}{3}}.$$

Furthermore, L and K are polynomial in λ and M , $L_{\text{sk}} = \lambda^{\frac{1}{3}} = \omega(\log \lambda)$, and $L - L_{\text{ct}} = \lambda^{\frac{1}{3}} = \omega(\log \lambda)$. Moreover, we have that $K \cdot L_{\text{sk}} = M^2\lambda^{\frac{4}{3}} \geq 2M^2\lambda = \kappa \cdot M \cdot L$ with $\kappa = 2$ (for all $\lambda \geq 8$). Thus, by [Theorem 4.9](#), [Construction 4.7](#) is unique-attribute-identity semantically secure for policies that achieve linear independence for unauthorized sets in the generic bilinear group model.

Lifting to full correctness and security. As discussed in [Remark 4.6](#), we can lift unique-attribute-identity semantic security to (standard) semantic security by having each authority use a PRF on the global identifier to derive key-generation randomness. Furthermore, [Remark 4.4](#) allows us to go from the inverse-polynomial error achieved by [Construction 4.7](#) to negligible correctness error. Applying these remarks with the above parameter instantiation, we obtain the following.

Corollary 4.19 (MA-ABE from Pairings in the BGGM). *Suppose one-way functions exist. Let λ be a security parameter, M be a policy-size bound, and let GroupGen be a prime-order asymmetric pairing-group generator. There exists a multi-authority attribute-based encryption scheme with global identifier space $\mathcal{G} = \{\mathcal{G}_\lambda\}_{\lambda \in \mathbb{N}}$ where $\mathcal{G}_\lambda = \{0, 1\}^\lambda$ and policy space $\mathcal{P}' = \{\mathcal{P}'_{M,\lambda}\}_{M,\lambda \in \mathbb{N}}$, where $\mathcal{P}'_{M,\lambda}$ consists of policies that can be described by a linear secret sharing scheme of size at most M that achieves linear independence for unauthorized sets, that is correct with negligible error and semantically secure in the generic bilinear group model.*

Acknowledgments

We thank George Lu and Mark Zhandry for helpful discussions. Brent Waters is supported by NSF CNS-2318701 and a Simons Investigator Award. David J. Wu is supported by NSF CNS-2140975, CNS-2318701, a Sloan Fellowship, a Microsoft Research Faculty Fellowship, a Google Research Scholar Award, an Amazon Research Award, and a gift from the Stellar Development Foundation.

Statement on AI use. All of the ideas in this paper were developed by the authors without the use of AI. After developing the main ideas (including the construction and the sequence of hybrids in the security analysis), the authors interacted with Claude to prepare a rough draft of the proofs and of the technical overview. The authors referenced this draft when preparing their own version of the technical exposition reflected in the current manuscript. The authors also used Claude for proofreading, copy-editing, and bug finding. The authors read over all AI-generated drafts and suggestions and made extensive edits to improve the exposition. The authors have verified the correctness and originality of all of the content in the paper.

References

- [ABD⁺26] Amit Agarwal, Kushal Babel, Sourav Das, Babak Poorebrahim Gilkalaye, Arup Mondal, Benny Pinkas, Peter Rindal, and Aayush Yadav. Weighted batched threshold encryption with applications to mempool privacy. In *IEEE S&P*, 2026.
- [ADG⁺26] Amit Agarwal, Sourav Das, Babak Poorebrahim Gilkalaye, Peter Rindal, and Victor Shoup. BTX: simple and efficient batch threshold encryption. *IACR Cryptol. ePrint Arch.*, 2026. Available at <https://eprint.iacr.org/2026/754>.
- [AFP25] Amit Agarwal, Rex Fernando, and Benny Pinkas. Efficiently-thresholdizable batched identity based encryption, with applications. In *CRYPTO*, 2025.
- [AGY26a] Abtin Afshar, Rishab Goyal, and Saikumar Yadugiri. Distributed monotone-policy encryption with silent setup from lattices. *IACR Cryptol. ePrint Arch.*, 2026. Available at <https://eprint.iacr.org/2026/372>.
- [AGY26b] Abtin Afshar, Rishab Goyal, and Saikumar Yadugiri. Silent distributed cryptography for DNFs and threshold policies from lattices. *IACR Cryptol. ePrint Arch.*, 2026. Available at <https://eprint.iacr.org/2026/1547>.
- [BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT*, 2004.
- [BB04b] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *EUROCRYPT*, 2004.
- [BCF⁺26] Jan Bormet, Arka Rai Choudhuri, Sebastian Faust, Sanjam Garg, Hussien Othman, Guru-Vamsi Policharla, Ziyang Qu, and Mingyuan Wang. BEAST-MEV: batched threshold encryption with silent setup for MEV prevention. In *ACM CCS*, 2026.
- [Bei96] Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Technion, 1996.
- [BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO*, 2001.
- [BFOQ25] Jan Bormet, Sebastian Faust, Hussien Othman, and Ziyang Qu. BEAT-MEV: epochless approach to batched threshold encryption for MEV prevention. In *USENIX Security Symposium*, 2025.
- [BL88] Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In *CRYPTO*, 1988.
- [BNRT26] Dan Boneh, Rohit Nema, Arnab Roy, and Ertem Nusret Tas. Efficient batch threshold encryption using partial fraction techniques. In *CRYPTO*, 2026.
- [BPR⁺08] Dan Boneh, Periklis Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *FOCS*, 2008.
- [BR06] Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. In *EUROCRYPT*, 2006.

- [BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *TCC*, 2011.
- [BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, 2007.
- [CGPW25] Arka Rai Choudhuri, Sanjam Garg, Guru-Vamsi Policharla, and Mingyuan Wang. Practical mempool privacy via one-time setup batched threshold encryption. In *USENIX Security Symposium*, 2025.
- [CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, 2003.
- [Coc01] Clifford C. Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding*, 2001.
- [CW26a] Jeffrey Champion and David J. Wu. Distributed monotone-policy encryption for DNFs from lattices. In *EUROCRYPT*, 2026.
- [CW26b] Jeffrey Champion and David J. Wu. Optimal distributed monotone-policy encryption for DNFs and more from lattices. *IACR Cryptol. ePrint Arch.*, 2026. Available at <https://eprint.iacr.org/2026/1464>.
- [DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 1976.
- [DKW21] Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for DNFs from LWE. In *EUROCRYPT*, 2021.
- [DKW23a] Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for NC^1 from BDH. *Journal of Cryptology*, 36(2), 2023.
- [DKW23b] Pratish Datta, Ilan Komargodski, and Brent Waters. Fully adaptive decentralized multi-authority ABE. In *EUROCRYPT*, 2023.
- [DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In *EUROCRYPT*, 2004.
- [DP23] Pratish Datta and Tapas Pal. Decentralized multi-authority attribute-based inner-product FE: large universe and unbounded. In *PKC*, 2023.
- [EHK⁺13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge L. Villar. An algebraic framework for Diffie-Hellman assumptions. In *CRYPTO*, 2013.
- [FPTX25] Rex Fernando, Guru-Vamsi Policharla, Andrei Tonkikh, and Zhuolun Xiang. TrX: encrypted mempools in high performance BFT protocols. *IACR Cryptol. ePrint Arch.*, 2025. Available at <https://eprint.iacr.org/2025/2032>.
- [Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, 2006.
- [GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS*, 2006.
- [GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *CRYPTO*, 2012.
- [GWWW26] Junqing Gong, Brent Waters, Hoeteck Wee, and David J. Wu. Threshold batched identity-based encryption from pairings in the plain model. In *EUROCRYPT*, 2026.
- [GY26] Rishab Goyal and Saikumar Yadugiri. Distributed monotone policy encryption with stronger security for DNFs and threshold policies from lattices. *IACR Cryptol. ePrint Arch.*, 2026. Available at <https://eprint.iacr.org/2026/1609>.

- [HK04] Swee-Huay Heng and Kaoru Kurosawa. k -resilient identity-based encryption in the standard model. In *CT-RSA*, 2004.
- [LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, 2010.
- [LW11] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In *EUROCRYPT*, 2011.
- [O’N10] Adam O’Neill. Definitional issues in functional encryption. *IACR Cryptol. ePrint Arch.*, 2010. Available at <https://eprint.iacr.org/2010/556>.
- [OT20] Tatsuki Okamoto and Katsuyuki Takashima. Decentralized attribute-based encryption and signatures. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 103-A(1), 2020.
- [Pol26] Guru-Vamsi Policharla. Labeled multi-key batched IBE. *IACR Cryptol. ePrint Arch.*, 2026. Available at <https://eprint.iacr.org/2026/1452>.
- [RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2), 1978.
- [RW15] Yannis Rouselakis and Brent Waters. Efficient statically-secure large-universe multi-authority attribute-based encryption. In *Financial Cryptography and Data Security*, 2015.
- [Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4), 1980.
- [Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, 1984.
- [Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, 1997.
- [SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In *ACM CCS*, 2010.
- [SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, 2005.
- [SWW26] Roy Stracovsky, Brent Waters, and David J. Wu. Pairing-based registered ABE for Boolean formulas with a linear-size CRS. In *CRYPTO*, 2026.
- [Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, 2005.
- [Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, 2009.
- [Zim15] Joe Zimmerman. How to obfuscate programs directly. In *EUROCRYPT*, 2015.
- [Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM*, 1979.
- [ZZ23] Cong Zhang and Mark Zhandry. The relationship between idealized models under computationally bounded adversaries. In *ASIACRYPT*, 2023.

A Identity-Based Encryption

As another illustration of our techniques, we describe an identity-based encryption (IBE) scheme from pairings with full security in the generic group model. Our scheme achieves collusion resistance through a different approach from existing pairing-based identity-based encryption schemes without random oracles (e.g., [CHK03, BB04a, Wat05, Gen06, Wat09, LW10]). Identity-based encryption provides a simpler demonstration of our core bounded- to full-collusion resistance transformation, which we extend in Sections 3 and 4 to build tagless batched identity-based encryption and multi-authority attribute-based encryption, respectively. As discussed in Section 1, we apply our approach to the plain-model variant [HK04] of the classic Boneh-Franklin IBE [BF01], where the random oracle is replaced by a K -wise independent hash function.

A.1 IBE Definitions

We start by recalling the syntax and security notions for identity-based encryption (IBE) [Sha84, BF01].

Definition A.1 (Identity-Based Encryption [Sha84, BF01, adapted]). An *identity-based encryption* (IBE) scheme with identity space $\mathcal{I} = \{\mathcal{I}_\lambda\}_{\lambda \in \mathbb{N}}$ is a tuple of efficient algorithms $\Pi_{\text{IBE}} = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ with the following syntax:

- $\text{Setup}(1^\lambda) \rightarrow (\text{mpk}, \text{msk})$: On input the security parameter λ , the setup algorithm outputs a master public key mpk and a master secret key msk . We assume that mpk implicitly contains the security parameter 1^λ and a description of the message space $\mathcal{M}_\lambda$ (where $|\mathcal{M}_\lambda| \geq 2$).
- $\text{KeyGen}(\text{msk}, \text{id}) \rightarrow \text{sk}_{\text{id}}$: On input the master secret key msk and an identity $\text{id} \in \mathcal{I}_\lambda$, the key-generation algorithm outputs an identity secret key sk_{id} .
- $\text{Encrypt}(\text{mpk}, \text{id}, m) \rightarrow \text{ct}$: On input the master public key mpk , an identity $\text{id} \in \mathcal{I}_\lambda$, and a message $m \in \mathcal{M}_\lambda$, the (randomized) encryption algorithm outputs a ciphertext ct .
- $\text{Decrypt}(\text{sk}_{\text{id}}, \text{ct}) \rightarrow m$: On input an identity secret key sk_{id} and a ciphertext ct , the decryption algorithm outputs a message m .

Definition A.2 (Correctness). An IBE scheme Π_{IBE} is *statistically correct* with error $\varepsilon(\lambda)$ if, for all $\lambda \in \mathbb{N}$, all master key-pairs (mpk, msk) in the support of $\text{Setup}(1^\lambda)$, all identities $\text{id} \in \mathcal{I}_\lambda$, all identity secret keys sk_{id} in the support of $\text{KeyGen}(\text{msk}, \text{id})$, and all messages $m \in \mathcal{M}_\lambda$ (where $\mathcal{M}_\lambda$ is the message space associated with mpk),

$$\Pr[\text{Decrypt}(\text{sk}_{\text{id}}, \text{ct}) = m : \text{ct} \leftarrow \text{Encrypt}(\text{mpk}, \text{id}, m)] \geq 1 - \varepsilon(\lambda).$$

Remark A.3 (Weaker Notions of Correctness). Like the batched IBE setting discussed in Remark 3.3, our definition above requires correctness to hold with a fixed probability for all master key-pairs and identity secret keys; that is, the probability is taken solely over the encryption randomness. This is the analog of “almost-all-keys perfect correctness” from [DNR04] (adapted to the setting of identity-based encryption). One could also consider weaker notions of correctness where this probability is taken over the randomness of key generation, or even both key generation and setup. While valid, these relaxations are meaningfully weaker. In particular, under our definition, we can reduce the decryption failure by encrypting the message multiple times and having the decrypter output the majority message. Under the weaker definitions, there could exist non-functional secret keys (or even a non-functional master key-pair) where decryption is guaranteed to fail irrespective of the encryption randomness.

Remark A.4 (Negligible Correctness Error). Like the batched IBE setting discussed in Remark 3.4, our definition allows the correctness error $\varepsilon(\lambda)$ to be non-negligible. When $\varepsilon(\lambda) \leq 1/2 - 1/\text{poly}(\lambda)$, we can achieve *negligible correctness error* by performing a $\text{poly}'(\lambda)$ -wise repetition of encryption (without performing multiple setups or generating multiple keys for each user) where $\text{poly}'(\lambda) = \omega(\log \lambda) \cdot \text{poly}(\lambda)^2$. The decryption algorithm then outputs the majority of the decrypted messages. This form of amplification is possible as the correctness error in Definition A.2 is taken over the encryption randomness (as opposed to the key-generation or setup randomness); see Remark A.3.

Definition A.5 (Semantic Security). For an IBE scheme Π_{IBE} , an adversary $\mathcal{A}$, a bit $b \in \{0, 1\}$, and a security parameter $\lambda \in \mathbb{N}$, we define the semantic security experiment as follows:

1. **Setup phase:** The challenger samples $(\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda)$ and gives mpk to $\mathcal{A}$.
2. **Pre-challenge query phase:** The adversary $\mathcal{A}$ can now make key-generation queries. On each query, the adversary specifies an identity $\text{id} \in \mathcal{I}_\lambda$ and the challenger replies with $\text{sk}_{\text{id}} \leftarrow \text{KeyGen}(\text{msk}, \text{id})$.
3. **Challenge phase:** At some point, $\mathcal{A}$ outputs a challenge identity $\text{id}^* \in \mathcal{I}_\lambda$ together with two challenge messages $m_0^*, m_1^* \in \mathcal{M}_\lambda$. The challenger replies with $\text{ct}^* \leftarrow \text{Encrypt}(\text{mpk}, \text{id}^*, m_b^*)$.
4. **Post-challenge query phase:** Now, $\mathcal{A}$ can continue to make key-generation queries. Like the pre-challenge query phase, the challenger replies with $\text{sk}_{\text{id}} \leftarrow \text{KeyGen}(\text{msk}, \text{id})$ when $\mathcal{A}$ queries an identity $\text{id} \in \mathcal{I}_\lambda$.

5. **Output phase:** At the end of the experiment, $\mathcal{A}$ outputs a bit $b^* \in \{0, 1\}$, which is the output of the experiment.

We say that an adversary $\mathcal{A}$ is admissible if it does not make key-generation queries on the challenge identity id^* . We furthermore say that Π_{IBE} is semantically secure if, for every efficient admissible adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$|\Pr[b^* = 1 : b = 0] - \Pr[b^* = 1 : b = 1]| \leq \text{negl}(\lambda).$$

Remark A.6 (Unique-Identity Security). [Definition A.5](#) allows an adversary to make several key-generation queries on a single identity, potentially receiving a different identity secret key on each query. We also consider security allowing only one key-generation query per identity, which we refer to as *unique-identity semantic security*. It is straightforward to upgrade unique-identity security to full security by using a pseudorandom function (PRF) to derive the key-generation randomness and then including the PRF key in the master secret key. Security of the composed construction follows by a standard hybrid argument (by first invoking PRF security and then invoking security of the underlying IBE scheme).

Remark A.7 (Bounded-Collusion Security). [Definition A.5](#) places *no a priori bound* on the number of key-generation queries: an efficient adversary may request keys for an arbitrary polynomial number of identities. An IBE scheme that is secure only when the number of queries is bounded by some fixed polynomial (fixed at setup time) is said to be *bounded-collusion resistant*. The goal of this work is to upgrade such schemes to full collusion resistance as defined above.

A.2 IBE Construction

In this section, we describe our construction of collusion-resistant identity-based encryption. As described in [Section 1.1](#), we obtain our scheme by augmenting the [\[HK04\]](#) variant of the Boneh-Franklin [\[BF01\]](#) IBE scheme (where the random oracle is replaced by a K -wise independent hash function) with our slot mechanism (see also [Fig. 1](#)). We give our full construction below:

Construction A.8 (Identity-Based Encryption). Let λ be a security parameter and define the slot count $L = L(\lambda)$, the number of slots $L_{\text{sk}} = L_{\text{sk}}(\lambda)$ associated with each secret key, number of slots $L_{\text{ct}} = L_{\text{ct}}(\lambda)$ associated with each ciphertext, and the collusion bound $K = K(\lambda)$. Let GroupGen be a prime-order asymmetric pairing-group generator. We construct an identity-based encryption scheme $\Pi_{\text{IBE}} = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ with identity space $\mathcal{I} = \{\mathcal{I}_\lambda\}_{\lambda \in \mathbb{N}}$ where $\mathcal{I}_\lambda = \{0, 1\}^\lambda$ as follows:

- **Setup(1^λ):** On input the security parameter λ , the setup algorithm proceeds as follows:
 - Sample a prime-order asymmetric pairing group $\bar{\mathcal{G}} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \bar{g}_1, \bar{g}_2, e) \leftarrow \text{GroupGen}(1^\lambda)$ and generators $g_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1 \setminus \{\bar{g}_1^0\}$ and $g_2 \xleftarrow{\mathbb{R}} \mathbb{G}_2 \setminus \{\bar{g}_2^0\}$. Define $[1]_1 := g_1$, $[1]_2 := g_2$, and $\mathcal{G} = (\bar{\mathcal{G}}, [1]_1, [1]_2)$.
 - Sample group elements $[u_1]_2, \dots, [u_K]_2 \xleftarrow{\mathbb{R}} \mathbb{G}_2$ and define the function $H(X) := \sum_{i \in [K]} X^{i-1} [u_i]_2$. Note that $H: \mathbb{Z}_p \rightarrow \mathbb{G}_2$ is a K -wise independent hash function keyed by $([u_1]_2, \dots, [u_K]_2)$.
 - Sample exponents $\alpha, \gamma_1, \dots, \gamma_L \xleftarrow{\mathbb{R}} \mathbb{Z}_p$.

Output the master public key $\text{mpk} = (\mathcal{G}, H, [\alpha]_1, \{[\gamma_i]_1\}_{i \in [L]})$ and master secret key $\text{msk} = (\mathcal{G}, H, \alpha, \{\gamma_i\}_{i \in [L]})$ and define the message space to be $\mathcal{M}_\lambda := \mathbb{G}_T$.

- **KeyGen(msk, id):** On input the master secret key $\text{msk} = (\mathcal{G}, H, \alpha, \{\gamma_i\}_{i \in [L]})$ and an identity $\text{id} \in \mathcal{I}_\lambda$, the key-generation algorithm proceeds as follows:
 - Sample a set $S_{\text{id}} \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$ and sample exponents $\beta_{i,\text{id}} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ for all $i \in S_{\text{id}}$.
 - Compute $[\delta_{\text{id}}]_2 = \alpha[h_{\text{id}}]_2 + \sum_{i \in S_{\text{id}}} [\gamma_i \beta_{i,\text{id}}]_2$ where $[h_{\text{id}}]_2 \leftarrow H(\text{id})$.

Output the identity secret key $\text{sk}_{\text{id}} = ([\delta_{\text{id}}]_2, \{[\beta_{i,\text{id}}]_2\}_{i \in S_{\text{id}}})$.

- **Encrypt($\text{mpk}, \text{id}, [m]_T$):** On input the master public key $\text{mpk} = (\mathcal{G}, H, [\alpha]_1, \{[\gamma_i]_1\}_{i \in [L]})$, an identity $\text{id} \in \mathcal{I}_\lambda$, and a message $[m]_T \in \mathcal{M}_\lambda$, the encryption algorithm proceeds as follows:

- Sample an exponent $s \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and compute the following ciphertext components:

$$[c_1]_{\mathbb{T}} = s[\alpha]_1 \cdot [h_{\text{id}}]_2 + [m]_{\mathbb{T}} \quad \text{and} \quad [c_2]_1 = [s]_1,$$

where $[h_{\text{id}}]_2 \leftarrow H(\text{id})$.

- Sample a set $S_{\text{ct}} \xleftarrow{\mathbb{R}} \{S \subseteq [L] : |S| = L_{\text{ct}}\}$ and let $[c_{3,i}]_1 = s[\gamma_i]_1$ for $i \in S_{\text{ct}}$ and $[c_{3,i}]_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1$ for $i \in [L] \setminus S_{\text{ct}}$.

Output the ciphertext $\text{ct} = ([c_1]_{\mathbb{T}}, [c_2]_1, \{[c_{3,i}]_1\}_{i \in [L]})$.

- **Decrypt**($\text{sk}_{\text{id}}, \text{ct}$): On input an identity secret key $\text{sk}_{\text{id}} = ([\delta_{\text{id}}]_2, \{[\beta_{i,\text{id}}]_2\}_{i \in S_{\text{id}}})$ and ciphertext $\text{ct} = ([c_1]_{\mathbb{T}}, [c_2]_1, \{[c_{3,i}]_1\}_{i \in [L]})$, the decryption algorithm outputs $[m]_{\mathbb{T}} = [c_1]_{\mathbb{T}} - [c_2]_1 \cdot [\delta_{\text{id}}]_2 + \sum_{i \in S_{\text{id}}} [c_{3,i}]_1 \cdot [\beta_{i,\text{id}}]_2$.

Theorem A.9 (Correctness). *Construction A.8 is statistically correct with error $\varepsilon(\lambda) = L_{\text{sk}}(1 - L_{\text{ct}}/L)$.*

Proof. Take any λ and any (mpk, msk) in the support of $\text{Setup}(1^\lambda)$ where $\text{mpk} = (\mathcal{G}, H, [\alpha]_1, \{[\gamma_i]_1\}_{i \in [L]})$. Next, take any identity $\text{id} \in \mathcal{I}_\lambda$ and any sk_{id} in the support of $\text{KeyGen}(\text{msk}, \text{id})$ where $\text{sk}_{\text{id}} = ([\delta_{\text{id}}]_2, \{[\beta_{i,\text{id}}]_2\}_{i \in S_{\text{id}}})$ for

$$[\delta_{\text{id}}]_2 = \alpha[h_{\text{id}}]_2 + \sum_{i \in S_{\text{id}}} [\gamma_i \beta_{i,\text{id}}]_2 = \left[\alpha h_{\text{id}} + \sum_{i \in S_{\text{id}}} \gamma_i \beta_{i,\text{id}} \right]_2.$$

Finally, for any $[m]_{\mathbb{T}} \in \mathcal{M}_\lambda$, let $\text{ct} \leftarrow \text{Encrypt}(\text{mpk}, \text{id}, [m]_{\mathbb{T}})$ where $\text{ct} = ([c_1]_{\mathbb{T}}, [c_2]_1, \{[c_{3,i}]_1\}_{i \in [L]})$ for

$$\begin{aligned} [c_1]_{\mathbb{T}} &= s[\alpha]_1 \cdot [h_{\text{id}}]_2 + [m]_{\mathbb{T}} = [s\alpha h_{\text{id}} + m]_{\mathbb{T}}, & [c_{3,i}]_1 &= \begin{cases} s[\gamma_i]_1 = [s\gamma_i]_1 & i \in S_{\text{ct}} \\ [r_i]_1 & i \notin S_{\text{ct}} \end{cases}, \\ [c_2]_1 &= [s]_1, \end{aligned}$$

where for each $i \notin S_{\text{ct}}$, $r_i \in \mathbb{Z}_p$ denotes the discrete logarithm of the uniformly random group element $[c_{3,i}]_1 \xleftarrow{\mathbb{R}} \mathbb{G}_1$ sampled by **Encrypt**. Now consider $[m^*]_{\mathbb{T}} \leftarrow \text{Decrypt}(\text{sk}_{\text{id}}, \text{ct})$. We have that

$$\begin{aligned} [m^*]_{\mathbb{T}} &= [c_1]_{\mathbb{T}} - [s]_1 \cdot [\delta_{\text{id}}]_2 + \sum_{i \in S_{\text{id}}} [c_{3,i}]_1 \cdot [\beta_{i,\text{id}}]_2 \\ &= [s\alpha h_{\text{id}} + m]_{\mathbb{T}} - \left[s\alpha h_{\text{id}} + s \sum_{i \in S_{\text{id}}} \gamma_i \beta_{i,\text{id}} \right]_{\mathbb{T}} + \sum_{i \in S_{\text{id}} \cap S_{\text{ct}}} [s\gamma_i \beta_{i,\text{id}}]_{\mathbb{T}} + \sum_{i \in S_{\text{id}} \setminus S_{\text{ct}}} [r_i \beta_{i,\text{id}}]_{\mathbb{T}} \\ &= \left[s\alpha h_{\text{id}} + m - s\alpha h_{\text{id}} - \sum_{i \in S_{\text{id}}} s\gamma_i \beta_{i,\text{id}} + \sum_{i \in S_{\text{id}} \cap S_{\text{ct}}} s\gamma_i \beta_{i,\text{id}} + \sum_{i \in S_{\text{id}} \setminus S_{\text{ct}}} r_i \beta_{i,\text{id}} \right]_{\mathbb{T}} \\ &= \left[m - \sum_{i \in S_{\text{id}} \setminus S_{\text{ct}}} s\gamma_i \beta_{i,\text{id}} + \sum_{i \in S_{\text{id}} \setminus S_{\text{ct}}} r_i \beta_{i,\text{id}} \right]_{\mathbb{T}} \\ &= \left[m + \sum_{i \in S_{\text{id}} \setminus S_{\text{ct}}} (r_i - s\gamma_i) \beta_{i,\text{id}} \right]_{\mathbb{T}}. \end{aligned}$$

Observe that if $S_{\text{id}} \subseteq S_{\text{ct}}$, then $S_{\text{id}} \setminus S_{\text{ct}} = \emptyset$ and consequently the blinding term $\sum_{i \in S_{\text{id}} \setminus S_{\text{ct}}} (r_i - s\gamma_i) \beta_{i,\text{id}} = 0$. Hence, in this event, $[m^*]_{\mathbb{T}} = [m]_{\mathbb{T}}$ and decryption succeeds. It remains to bound the probability that $S_{\text{id}} \not\subseteq S_{\text{ct}}$. Since S_{ct} is sampled independently of S_{id} , we can appeal to a union bound and conclude

$$\Pr[S_{\text{id}} \not\subseteq S_{\text{ct}}] \leq \sum_{i \in S_{\text{id}}} \Pr[i \notin S_{\text{ct}}] = L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L} \right). \quad \square$$

Theorem A.10 (Security). *Suppose $2^\lambda < p = 2^{\Theta(\lambda)}$, $L = \text{poly}_1(\lambda)$ and $K = \text{poly}_2(\lambda)$, $L_{\text{sk}} = \omega(\log \lambda)$ and $K \cdot L_{\text{sk}} \geq \kappa \cdot L$ for some constant $\kappa \geq 2$, and $L - L_{\text{ct}} = \omega(\log \lambda)$. Then Construction A.8 is unique-identity semantically secure in the generic bilinear group model of order p .*

Proof. Consider the following hybrids parameterized by a bit b and a security parameter λ and, implicitly, an efficient admissible adversary $\mathcal{A}$. We use **green** to denote changes between hybrids.

- **Hybrid $\text{Hyb}_0^{(b)}(\lambda)$:** This hybrid is equivalent to an admissible adversary $\mathcal{A}$ playing the unique-identity semantic security game against [Construction A.8](#) where the underlying bilinear group is modeled as a *generic* bilinear group of order p as per [Definition 2.2](#). We let q_{KG} , q_{BGGM} , and q_{ZT} denote upper bounds on the respective number of key-generation, generic bilinear group operation (i.e., encode, add, and pair oracles), and zero-test oracle queries $\mathcal{A}$ makes. The game proceeds as follows:

- **Setup phase:** The challenger performs the setup procedure as follows:

- * The challenger samples a random prime $2^\lambda < p = 2^{\Theta(\lambda)}$, sets the BGGM handle space $\mathcal{H} = \{0, 1\}^\lambda$, and also initializes the table $\mathcal{T}$ which maps handles to exponents.
- * Next, it samples exponents $u_1, \dots, u_K \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and executes $\text{BGGM.Encode}(u_i, 2)$ to produce a handle $\langle u_i \rangle$ for each $i \in [K]$. Define the hash function $H'(X) \coloneqq \sum_{i \in [K]} X^{i-1} u_i$.
- * Finally, it samples exponents $\alpha, \gamma_1, \dots, \gamma_L \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and executes $\text{BGGM.Encode}(\alpha, 1)$ to produce a handle $\langle \alpha \rangle$ and, for each $i \in [L]$, $\text{BGGM.Encode}(\gamma_i, 1)$ to produce a handle $\langle \gamma_i \rangle$.

The challenger sends the master public key $\text{mpk} = (p, \{\langle u_i \rangle\}_{i \in [K]}, \langle \alpha \rangle, \{\langle \gamma_i \rangle\}_{i \in [L]})$ to the adversary $\mathcal{A}$.

- **Pre-challenge key-generation phase:** The adversary $\mathcal{A}$ may make key-generation queries on *distinct* identities of its choosing (as we are working in the unique-identity setting). Let id_ξ be the identity queried on the ξ^{th} key-generation query. The challenger generates an identity secret key as follows:

- * The challenger samples a set $S_\xi \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$.
- * Next, for each $i \in S_\xi$, it samples an exponent $\beta_{i,\xi} \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ and executes $\text{BGGM.Encode}(\beta_{i,\xi}, 2)$ which produces a handle $\langle \beta_{i,\xi} \rangle$.
- * Finally, it computes $\delta_\xi = \alpha h_\xi + \sum_{i \in S_\xi} \gamma_i \beta_{i,\xi}$ where $h_\xi \leftarrow H'(\text{id}_\xi)$. It then executes $\text{BGGM.Encode}(\delta_\xi, 2)$ which produces a handle $\langle \delta_\xi \rangle$.

The challenger sends the identity secret key $\text{sk}_\xi = (\langle \delta_\xi \rangle, \{\langle \beta_{i,\xi} \rangle\}_{i \in S_\xi})$ to the adversary $\mathcal{A}$. During this phase, $\mathcal{A}$ may also query the generic group oracles. We reiterate the oracles and their functionality below:

- * $\text{BGGM.Encode}(x, v)$: On input an exponent $x \in \mathbb{Z}_p$ and an index $v \in \{1, 2, T\}$, the oracle samples a handle $\langle h \rangle \xleftarrow{\mathbb{R}} \mathcal{H}$ and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- * $\text{BGGM.Add}(\langle h_1 \rangle, \langle h_2 \rangle)$: On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if $v_1 \neq v_2$. It then computes $x = x_1 + x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{\mathbb{R}} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, v_1)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- * $\text{BGGM.Pair}(\langle h_1 \rangle, \langle h_2 \rangle)$: On input handles $\langle h_1 \rangle, \langle h_2 \rangle \in \mathcal{H}$, the oracle retrieves the entries $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$ and aborts with output “invalid” if the entries are undefined or if either $v_1 \neq 1$ or $v_2 \neq 2$. It then computes $x = x_1 x_2 \in \mathbb{Z}_p$, samples a handle $\langle h \rangle \xleftarrow{\mathbb{R}} \mathcal{H}$, and aborts with output “collision” if $\mathcal{T}(\langle h \rangle)$ is already defined. Otherwise, it adds the entry $\langle h \rangle \mapsto (x, T)$ to $\mathcal{T}$ and returns $\langle h \rangle$.
- * $\text{BGGM.ZeroTest}(\langle h \rangle)$: On input a handle $\langle h \rangle \in \mathcal{H}$, the oracle retrieves $\mathcal{T}(\langle h \rangle) = (x, v)$ and aborts with output “invalid” if the entry is undefined. It returns “zero” if $x = 0 \in \mathbb{Z}_p$ and “non-zero” otherwise.
- **Challenge phase:** In this phase, the adversary $\mathcal{A}$ outputs a challenge identity id^* (which was not queried during the pre-challenge phase as $\mathcal{A}$ is admissible) together with handles $\langle m_0^* \rangle, \langle m_1^* \rangle$ for $\mathcal{A}$ ’s challenge messages such that $\mathcal{T}(\langle m_0^* \rangle) = (m_0^*, T)$ and $\mathcal{T}(\langle m_1^* \rangle) = (m_1^*, T)$. Namely, both messages are in the target group. The challenger produces the challenge ciphertext ct^* encrypting $\langle m_b^* \rangle$ under id^* as follows:
 - * The challenger samples an exponent $s \xleftarrow{\mathbb{R}} \mathbb{Z}_p$, retrieves the entry $\mathcal{T}(\langle m_b^* \rangle) = (m_b^*, T)$, and then runs $\text{BGGM.Encode}(s h^* + m_b^*, T)$ for $h^* \leftarrow H'(\text{id}^*)$ to produce a handle $\langle c_1^* \rangle$.
 - * Next, it executes $\text{BGGM.Encode}(s, 1)$ to produce a handle $\langle c_2^* \rangle$.
 - * Finally, it samples a set $S^* \xleftarrow{\mathbb{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. For each $i \in [L]$, it computes $c_{3,i}^* = s \gamma_i$ if $i \in S^*$ or samples $c_{3,i}^* \xleftarrow{\mathbb{R}} \mathbb{Z}_p$ otherwise, and it executes $\text{BGGM.Encode}(c_{3,i}^*, 1)$ to produce a handle $\langle c_{3,i}^* \rangle$.

The challenger sends the challenge ciphertext $\text{ct}^* = (\langle c_1^* \rangle, \langle c_2^* \rangle, \{\langle c_{3,i}^* \rangle\}_{i \in [L]})$ to the adversary $\mathcal{A}$.

- **Post-challenge key-generation phase:** The (admissible) adversary $\mathcal{A}$ may again make key-generation queries on *distinct* identities of its choosing provided they are not the challenge identity id^* nor were they previously queried in the pre-challenge phase. The challenger generates identity secret keys using the same procedure as the pre-challenge key-generation phase. During this phase, $\mathcal{A}$ may again make encode, add, pair, and zero-test oracle queries as specified above.
- **Output phase:** The adversary outputs a bit b^* which is the output of the game.
- **Hybrid $\text{Hyb}_1^{(b)}(\lambda)$:** This hybrid is equivalent to hybrid $\text{Hyb}_0^{(b)}$, except we add the following halting condition:
 - **Key-generation queries:** If there exists a collection of K sets $S_{\xi_1}, \dots, S_{\xi_K} \subseteq [L]$ sampled during distinct key-generation queries (i.e., $\xi_1, \dots, \xi_K$ are all different) such that $|\bigcup_{i \in [K]} S_{\xi_i}| \leq L/\kappa$, halt with output 0.
- **Hybrid $\text{Hyb}_2^{(b)}(\lambda)$:** This hybrid is equivalent to hybrid $\text{Hyb}_1^{(b)}$, except we add the following halting condition:
 - **Generic group oracle queries:** If BGGM.Encode , BGGM.Add , or BGGM.Pair returns “collision” because the sampled handle $\langle h \rangle$ has previously been sampled (i.e., $\mathcal{T}(\langle h \rangle)$ is already defined), halt with output 0.
- **Hybrid $\text{Hyb}_3^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_2^{(b)}$ (including its handle collision-abort rule) except that it augments the challenger so that, in addition to maintaining handles that refer to group elements, it also keeps track of two kinds of formal polynomials to which the handles map. Specifically, handles given to the adversary map to polynomials over a set of formal variables $\hat{\mathcal{V}}$, while the challenger simultaneously maintains polynomials over a set of formal variables $\tilde{\mathcal{V}}$ that it will use to evaluate zero-test queries. We refer to the former as the *adversary’s* formal polynomials and the latter as the *challenger’s* formal polynomials. To distinguish the representations used below, a plain symbol x denotes an actual exponent in $\mathbb{Z}_p$, $[x]_v$ denotes the corresponding concrete group element in group $\mathbb{G}_v$, and $\langle x \rangle$ denotes an opaque handle for that group element in the generic-group game. We use a hat (e.g., $\hat{p}$) to denote polynomials using only the components exposed to the adversary and a tilde (e.g., $\tilde{p}$) to denote the same polynomial expanded in terms of the challenger’s underlying sampled values.

The set of variables $\hat{\mathcal{V}}$ over which the adversary’s formal polynomials are defined captures the components the construction returns during setup, key generation, and encryption. In detail, $\hat{\mathcal{V}}$ is specified as follows:

$$\begin{aligned}\hat{\mathcal{V}}_1 &= \{\hat{\alpha}, \hat{c}_2^*\} \cup \{\hat{y}_i, \hat{c}_{3,i}^*\}_{i \in [L]}, \\ \hat{\mathcal{V}}_2 &= \{\hat{u}_i\}_{i \in [K]} \cup \{\hat{\delta}_\xi\}_{\xi \in [q_{\text{KG}}]} \cup \{\hat{\beta}_{i,\xi}\}_{(i,\xi) \in [L] \times [q_{\text{KG}}]}, \\ \hat{\mathcal{V}}_T &= \{\hat{c}_1^*\}, \\ \hat{\mathcal{V}} &= \hat{\mathcal{V}}_1 \cup \hat{\mathcal{V}}_2 \cup \hat{\mathcal{V}}_T.\end{aligned}$$

Thus, $\hat{\mathcal{V}}$ contains variables only for handles supplied directly by the challenger. Handles produced by the pairing are represented by degree-two monomials in these variables. Consequently, a formal polynomial associated with a handle in $\mathbb{G}_T$ lies in the linear span of $\{1\} \cup \hat{\mathcal{V}}_1 \cup \hat{\mathcal{V}}_2 \cup \hat{\mathcal{V}}_T \cup \{\hat{v}_1 \hat{v}_2 : (\hat{v}_1, \hat{v}_2) \in \hat{\mathcal{V}}_1 \times \hat{\mathcal{V}}_2\}$.

The set of variables $\tilde{\mathcal{V}}$ over which the challenger’s formal polynomials are defined captures the group elements sampled when running setup, key generation, and encryption. In detail, $\tilde{\mathcal{V}}$ is specified as follows:

$$\tilde{\mathcal{V}} = \{\tilde{\alpha}, \tilde{s}\} \cup \{\tilde{u}_i\}_{i \in [K]} \cup \{\tilde{y}_i, \tilde{r}_i\}_{i \in [L]} \cup \{\tilde{\beta}_{i,\xi}\}_{(i,\xi) \in [L] \times [q_{\text{KG}}]}.$$

At any time t in the hybrid, define an assignment $\phi_t : \tilde{\mathcal{V}} \rightarrow \mathbb{Z}_p$ that maps each challenger variable $\tilde{v} \in \tilde{\mathcal{V}}$ to the corresponding value $v \in \mathbb{Z}_p$ if it has been sampled by time t and to 0 otherwise. We let $\text{Eval}_{\phi_t} : \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p$ denote the standard polynomial-evaluation ring homomorphism induced by ϕ_t . Thus, in particular, the variables $\tilde{r}_i$ and $\tilde{\beta}_{i,\xi}$ evaluate to 0 until the corresponding values have been sampled.

With adversary and challenger formal polynomials defined, the hybrid proceeds as follows:

- **Setup phase:** The challenger performs the setup procedure with the following additional steps:
 - * When the challenger initializes the table $\mathcal{T}$, it additionally initializes tables $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$.
 - * With handles $\langle u_i \rangle$ for each $i \in [K]$, it adds the entries $\langle u_i \rangle \mapsto \hat{u}_i$ to $\hat{\mathcal{T}}$ and $\langle u_i \rangle \mapsto \tilde{u}_i$ to $\tilde{\mathcal{T}}$.
 - * Finally, when it produces handles $\langle \alpha \rangle, \langle \gamma_1 \rangle, \dots, \langle \gamma_L \rangle$, it adds the entries $\langle \alpha \rangle \mapsto \hat{\alpha}$ and $\langle \gamma_i \rangle \mapsto \hat{\gamma}_i$ for each $i \in [L]$ to $\hat{\mathcal{T}}$, and it also adds the entries $\langle \alpha \rangle \mapsto \tilde{\alpha}$ and $\langle \gamma_i \rangle \mapsto \tilde{\gamma}_i$ for each $i \in [L]$ to $\tilde{\mathcal{T}}$.
- **Pre-challenge key-generation phase:** On the ξ^{th} key-generation query on identity id_ξ , the challenger generates an identity secret key with the following additional steps:
 - * When the challenger produces handles $\langle \beta_{i,\xi} \rangle$ for each $i \in S_\xi$, it adds the entries $\langle \beta_{i,\xi} \rangle \mapsto \hat{\beta}_{i,\xi}$ to $\hat{\mathcal{T}}$ and $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ to $\tilde{\mathcal{T}}$.
 - * When it computes $\delta_\xi = \alpha h_\xi + \sum_{i \in S_\xi} \gamma_i \beta_{i,\xi}$ to produce the handle $\langle \delta_\xi \rangle$, it adds the entries $\langle \delta_\xi \rangle \mapsto \hat{\delta}_\xi$ to $\hat{\mathcal{T}}$ and $\langle \delta_\xi \rangle \mapsto \tilde{\delta}_\xi$ to $\tilde{\mathcal{T}}$. Recall that $h_\xi = \sum_{i \in [K]} \text{id}_\xi^{i-1} u_i$.

When $\mathcal{A}$ makes generic group oracle queries, the challenger performs the following additional steps:

- * **BGGM.Encode(x, v):** In addition to adding the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger adds the entry $\langle h \rangle \mapsto x$ to both $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ where x denotes the constant polynomial.
- * **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** When it adds the entry $\langle h \rangle \mapsto (x_1 + x_2, v_1)$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger additionally adds the entries $\langle h \rangle \mapsto \hat{\mathcal{T}}(\langle h_1 \rangle) + \hat{\mathcal{T}}(\langle h_2 \rangle)$ to $\hat{\mathcal{T}}$ and $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$, where both sums denote formal polynomial addition.
- * **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** When it adds the entry $\langle h \rangle \mapsto (x_1 x_2, T)$ to $\mathcal{T}$ (i.e., no halt occurred), the challenger additionally adds the entries $\langle h \rangle \mapsto \hat{\mathcal{T}}(\langle h_1 \rangle) \cdot \hat{\mathcal{T}}(\langle h_2 \rangle)$ to $\hat{\mathcal{T}}$ and $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$, where both products denote multiplication in the formal polynomial ring.
- * **BGGM.ZeroTest($\langle h \rangle$):** If $\langle h \rangle$ is present in $\mathcal{T}$, then it returns “zero” if

$$\text{Eval}_{\phi_t}(\tilde{h}) = 0$$

for the challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ at the time t of the query, and returns “non-zero” otherwise.

- **Challenge phase:** This phase includes the following additional steps:
 - * When the challenger produces the handle $\langle c_1^* \rangle$, it also adds the entries $\langle c_1^* \rangle \mapsto \hat{c}_1^*$ to $\hat{\mathcal{T}}$ and $\langle c_1^* \rangle \mapsto \tilde{s} \tilde{\alpha} \sum_{i \in [K]} \text{id}^{*i-1} \tilde{u}_i + \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ to $\tilde{\mathcal{T}}$ where $\langle m_b^* \rangle$ is the handle for the challenge message.
 - * With the handle $\langle c_2^* \rangle$, it adds the entries $\langle c_2^* \rangle \mapsto \hat{c}_2^*$ to $\hat{\mathcal{T}}$ and $\langle c_2^* \rangle \mapsto \tilde{s}$ to $\tilde{\mathcal{T}}$.
 - * Finally, for the handles $\langle c_{3,i}^* \rangle$ for each $i \in [L]$, it adds the entry $\langle c_{3,i}^* \rangle \mapsto \hat{c}_{3,i}^*$ to $\hat{\mathcal{T}}$. It additionally adds to $\tilde{\mathcal{T}}$ the entry $\langle c_{3,i}^* \rangle \mapsto \tilde{s} \tilde{\gamma}_i$ if $i \in S^*$ and $\langle c_{3,i}^* \rangle \mapsto \tilde{r}_i$ otherwise.
- **Post-challenge key-generation phase:** This phase is the same as the pre-challenge key-generation phase. In particular, at the time t of each zero-test query, the challenger uses the current assignment ϕ_t , which now includes the values sampled during the challenge phase, and returns “zero” if $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ (where $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ is the challenger formal polynomial associated with the handle $\langle h \rangle$ queried) and “non-zero” otherwise.

- **Hybrid $\text{Hyb}_4^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_3^{(b)}$, except for the following change:

- **Zero-test oracle queries:** When the adversary $\mathcal{A}$ queries a handle $\langle h \rangle$ to the zero-test oracle, the challenger no longer evaluates the associated challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ under the current assignment ϕ_t . Instead, it returns “zero” if $\tilde{\mathcal{T}}(\langle h \rangle) \equiv 0$ in $\mathbb{Z}_p[\tilde{\mathcal{V}}]$ and “non-zero” otherwise. That is, this hybrid tests whether $\tilde{\mathcal{T}}(\langle h \rangle)$ is identically zero as a formal polynomial, while previous hybrids tested whether $\text{Eval}_{\phi_t}(\tilde{\mathcal{T}}(\langle h \rangle)) = 0$.

- **Hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}(\lambda)$:** This hybrid is equivalent to $\text{Hyb}_4^{(b)}$, except for the following change:

- **Challenge phase:** For each handle $\langle c_{3,i}^* \rangle$ for $i \in [L]$, the challenger always adds to $\tilde{\mathcal{T}}$ the entry $\langle c_{3,i}^* \rangle \mapsto \tilde{r}_i$.⁵ Previously, it sampled a set S^* and added the entry $\langle c_{3,i}^* \rangle \mapsto \tilde{s}_{\tilde{y}_i}$ to $\tilde{\mathcal{T}}$ if $i \in S^*$ and $\langle c_{3,i}^* \rangle \mapsto \tilde{r}_i$ to $\tilde{\mathcal{T}}$ otherwise.
- **Post-challenge query phase:** At the start of the post-challenge query phase, the challenger does the following additional steps:
 - * The challenger samples a set $S^* \xleftarrow{\mathcal{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$ and defines $\text{Sub}_{S^*} : \mathbb{Z}_p[\tilde{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ to be the ring homomorphism induced by

$$\tilde{r}_i \mapsto \tilde{s}_{\tilde{y}_i} \quad \text{for } i \in S^* \quad \text{and} \quad \tilde{v} \mapsto \tilde{v} \quad \text{otherwise.}$$

For each handle $\langle h \rangle$ in $\tilde{\mathcal{T}}$, let $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ be the challenger's formal polynomial indexed by $\langle h \rangle$ and define $\tilde{h}' = \text{Sub}_{S^*}(\tilde{h})$ (i.e., the formal challenger polynomial $\tilde{h}'$ is $\tilde{h}$ where the $\tilde{r}_i$ components are substituted with $\tilde{s}_{\tilde{y}_i}$ for $i \in S^*$). Replace the entry $\langle h \rangle \mapsto \tilde{h}$ in $\tilde{\mathcal{T}}$ with $\langle h \rangle \mapsto \tilde{h}'$.

- **Hybrid $\text{Hyb}_{5,\chi}^{(b)}(\lambda)$ for $\chi \in [q_{\text{ZT}}]$:** This hybrid is equivalent to $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$, except for the following change:
 - **Post-challenge zero-test oracle queries:** On the χ^{th} post-challenge query to the zero-test oracle, the challenger does the following based on the adversary's formal polynomial $\hat{h} = \hat{\mathcal{T}}(\langle h \rangle)$ associated with the handle $\langle h \rangle$ queried by the adversary:
 - * **Message-dependent component:** If $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$ (i.e., there is a $\hat{c}_1^*$ term in the adversary's formal polynomial associated with $\langle h \rangle$), then the oracle returns “non-zero” (regardless of whether it is identical to the zero polynomial or not).
 - * **Challenge-set-dependent component:** If $\text{coeff}_{\hat{c}_{3,i}^* \hat{v}}(\hat{h}) \neq 0$ for some $i \in [L]$ and $\hat{v} \in \{1\} \cup \hat{\mathcal{V}}_2$, then the oracle returns “non-zero” (regardless of whether it is identical to the zero polynomial or not).
 - * If neither of the cases above occur, and for the remaining queries to the zero-test oracle, the challenger proceeds as in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$.

Note that because the hybrid is defined recursively, all post-challenge zero-test oracle queries before the χ^{th} query have already also been changed to include the above behavior.

- **Hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}(\lambda)$ for $\chi \in [q_{\text{ZT}}]$:** This hybrid is equivalent to $\text{Hyb}_{5,\chi}^{(b)}$, except for the following change:
 - **Post-challenge zero-test oracle queries:** After the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query, the challenger no longer samples S^* and performs the Sub_{S^*} substitution. Instead, after the χ^{th} post-challenge query to the zero-test oracle, the challenger performs this substitution, explicitly doing the following:
 - * Sample a set $S^* \xleftarrow{\mathcal{R}} \{S \in 2^{[L]} : |S| = L_{\text{ct}}\}$. For each handle $\langle h \rangle$ in $\tilde{\mathcal{T}}$, let $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ be the challenger's formal polynomial indexed by $\langle h \rangle$ and let $\tilde{h}' = \text{Sub}_{S^*}(\tilde{h})$ (i.e., $\tilde{h}'$ is the polynomial $\tilde{h}$ where the $\tilde{r}_i$ components are substituted with $\tilde{s}_{\tilde{y}_i}$ for $i \in S^*$). Replace the entry $\langle h \rangle \mapsto \tilde{h}$ in $\tilde{\mathcal{T}}$ with the entry $\langle h \rangle \mapsto \tilde{h}'$.
- In the previous hybrid, the step was performed after the first $\chi - 1$ post-challenge queries to the zero-test oracle, instead of the χ^{th} post-challenge query.

Lemma A.11. For $L = \text{poly}_1(\lambda)$ and $K = \text{poly}_2(\lambda)$, $L_{\text{sk}} = \omega(\log \lambda)$ and $K \cdot L_{\text{sk}} \geq \kappa \cdot L$ for some constant $\kappa \geq 2$, there exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,

$$|\Pr[\text{Hyb}_0^{(b)}(\lambda)] - \Pr[\text{Hyb}_1^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The hybrids are equivalent except for the event that the halting condition in $\text{Hyb}_1^{(b)}$ is triggered. Namely, the experiments are equivalent unless there exists some collection of sets $S_{\xi_1}, \dots, S_{\xi_K} \subseteq [L]$ sampled during (pre- or post-challenge) key-generation for K different keys $\xi_1, \dots, \xi_K$ such that the union $|\bigcup_{i \in [K]} S_{\xi_i}| \leq L/\kappa$, or equivalently, unless there exists some set $T \subseteq [L]$ with $|T| = L/\kappa$ such that $S_{\xi_i} \subseteq T$ for all $i \in [K]$. Let bad denote this event.

⁵Note that the behavior of the hybrid on actual exponents is irrelevant to the security analysis, as nothing computed from these exponents is given to the adversary in this hybrid or later. Thus, for simplicity, we elide this detail in the specification of the hybrid.

We first consider the probability that a given set $S_{\xi_j} \xleftarrow{R} \{S \in 2^{[L]} : |S| = L_{\text{sk}}\}$ is contained within a *fixed* set T of size L/κ . We have that

$$\Pr[S_{\xi_j} \subseteq T] \leq \binom{L/\kappa}{L_{\text{sk}}} \cdot \binom{L}{L_{\text{sk}}}^{-1} \leq \left(\frac{L/\kappa}{L}\right)^{L_{\text{sk}}} = \kappa^{-L_{\text{sk}}}.$$

Now, because each set $S_{\xi_1}, \dots, S_{\xi_K}$ is sampled independently of each other (and any queries the adversary does, as each key-generation query samples a fresh set), we have that

$$\Pr[\forall i \in [K] : S_{\xi_i} \subseteq T] = \Pr[S_{\xi_j} \subseteq T]^K \leq \kappa^{-K \cdot L_{\text{sk}}}.$$

To analyze the full bad event, observe that there are $\binom{q_{\text{KG}}}{K}$ possible collections of K sets and $\binom{L}{L/\kappa}$ possible target sets T . By applying the union bound and using the fact that $\kappa \geq 2$, we have that,

$$\begin{aligned} \Pr[\text{bad}] &\leq \binom{q_{\text{KG}}}{K} \cdot \binom{L}{L/\kappa} \cdot \Pr[\forall i \in [K] : S_{\xi_i} \subseteq T] \\ &\leq \left(\frac{e \cdot q_{\text{KG}}}{K}\right)^K \cdot \kappa^L \cdot \kappa^{-K \cdot L_{\text{sk}}} \\ &\leq \kappa^{K \cdot \log(e \cdot q_{\text{KG}}/K) / \log \kappa + L - K \cdot L_{\text{sk}}}. \end{aligned}$$

Finally, recall that $L = \text{poly}_1(\lambda) \leq \lambda^{c_1}$ for all $\lambda \geq d_1$, $K = \text{poly}_2(\lambda) \leq \lambda^{c_2}$ for all $\lambda \geq d_2$, and $q_{\text{KG}} = \text{poly}_3(\lambda) \leq \lambda^{c_3}$ for all $\lambda \geq d_3$ for some $c_1, c_2, c_3, d_1, d_2, d_3 \in \mathbb{N}$, and that $K \cdot L_{\text{sk}} \geq \kappa \cdot L$. Consider any $c_4 > 0$. Because $L_{\text{sk}} = \omega(\log \lambda)$, there exists some $d_4 \in \mathbb{N}$ such that $L_{\text{sk}} \geq c_4 \log \lambda$ for all $\lambda \geq d_4$. Let $c = (1 - 1/\kappa) \log \kappa \cdot c_4 - c_3 - 1$ and $d = \max(d_3, d_4, 3)$. We have that,

$$\begin{aligned} \Pr[\text{bad}] &\leq \kappa^{K \cdot \log(e \cdot \lambda^{c_3}) / \log \kappa + K \cdot L_{\text{sk}} / \kappa - K \cdot L_{\text{sk}}} && \text{for } \lambda \geq d_3 \\ &\leq \kappa^{K \cdot ((c_3+1) \log \lambda / \log \kappa - (1-1/\kappa) \cdot c_4 \log \lambda)} && \text{for } \lambda \geq d \\ &= \lambda^{-K \cdot c} && \text{for } \lambda \geq d \\ &\leq \lambda^{-c} && \text{for } \lambda \geq d, \text{ if } c \geq 0. \end{aligned}$$

Note that since $c_4 > 0$ was arbitrary, c can take any positive value. That is, for every $c > 0$, it holds that $\Pr[\text{bad}] \leq \lambda^{-c}$ for all sufficiently large λ , and so there exists some negligible function $\text{negl}(\cdot)$ such that

$$|\Pr[\text{Hyb}_0^{(b)}(\lambda)] - \Pr[\text{Hyb}_1^{(b)}(\lambda)]| \leq \text{negl}(\lambda). \quad \square$$

Lemma A.12. For $L = \text{poly}_1(\lambda)$ and $K = \text{poly}_2(\lambda)$, there exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,

$$|\Pr[\text{Hyb}_1^{(b)}(\lambda)] - \Pr[\text{Hyb}_2^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The two hybrids are identical except for the event that the halting condition in $\text{Hyb}_2^{(b)}$ is triggered (i.e., one of the handles sampled during the experiment is equal to a previously sampled handle). Let bad denote this event. Since each handle is sampled uniformly from $\mathcal{H} = \{0, 1\}^\lambda$, a union bound over pairs of sampled handles over the course of the hybrid (both sampled during adversarial queries and by the challenger) gives us

$$\Pr[\text{bad}] \leq 2^{-\lambda} \cdot \left(\underbrace{K + L + 1}_{\text{setup phase}} + \underbrace{q_{\text{KG}} \cdot (L_{\text{sk}} + 1)}_{\text{key generation queries}} + \underbrace{q_{\text{BGGM}}}_{\text{generic group queries}} + \underbrace{L + 3}_{\text{challenge phase}} \right)^2.$$

Finally, recall that $L = \text{poly}_1(\lambda) \leq \lambda^{c_1}$ for all $\lambda \geq d_1$, $K = \text{poly}_2(\lambda) \leq \lambda^{c_2}$ for all $\lambda \geq d_2$, $q_{\text{KG}} = \text{poly}_3(\lambda) \leq \lambda^{c_3}$ for all $\lambda \geq d_3$, and $q_{\text{BGGM}} = \text{poly}_4(\lambda) \leq \lambda^{c_4}$ for all $\lambda \geq d_4$ for some $c_1, c_2, c_3, c_4, d_1, d_2, d_3, d_4 \in \mathbb{N}$. Furthermore, $L_{\text{sk}} \leq L$. Let

$c = \max(c_1, c_2, c_3, c_4)$ and $d = \max(d_1, d_2, d_3, d_4)$. We have that

$$\begin{aligned}
\Pr[\text{bad}] &\leq 2^{-\lambda} \cdot \left(\lambda^{c_2} + \lambda^{c_1} + 1 + \lambda^{c_3} (\lambda^{c_1} + 1) + \lambda^{c_4} + \lambda^{c_1} + 3 \right)^2 && \text{for } \lambda \geq d \\
&\leq 2^{-\lambda} \cdot \left(\lambda^{2c} + 5\lambda^c + 4 \right)^2 && \text{for } \lambda \geq d \\
&\leq 2^{-\lambda} \cdot (16\lambda^{2c})^2 && \text{for } \lambda \geq d \\
&= 2^{8-\lambda+4c \log \lambda} && \text{for } \lambda \geq d \\
&\leq 2^{-\frac{1}{2}\lambda} && \text{for } \lambda \geq \max(d, 256c^2, 32).
\end{aligned}$$

It follows that there exists some negligible function $\text{negl}(\cdot)$ such that

$$|\Pr[\text{Hyb}_1^{(b)}(\lambda)] - \Pr[\text{Hyb}_2^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

□

Lemma A.13. For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_2^{(b)}(\lambda)] = \Pr[\text{Hyb}_3^{(b)}(\lambda)]$.

Proof. The two hybrids are identical except that, in hybrid $\text{Hyb}_3^{(b)}$, the challenger additionally maintains adversary and challenger formal polynomial tables $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{T}}^*$ respectively, and it answers a zero-test query on a handle $\langle h \rangle$ at time t by testing whether $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ for the challenger's formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$. In hybrid $\text{Hyb}_2^{(b)}$, the challenger instead tested whether $x = 0$ for the exponent x where $\mathcal{T}(\langle h \rangle) = (x, v)$. Because neither table is given to the adversary $\mathcal{A}$ and neither is used to answer any other oracle query, we need only argue that these two ways of answering a zero-test query always agree. We do so by showing that the challenger's formal polynomial $\tilde{h}$ associated with a handle $\langle h \rangle$ always evaluates to the exponent x held in $\mathcal{T}(\langle h \rangle)$.

Before continuing with the analysis, we first recall the entries the challenger adds to $\tilde{\mathcal{T}}$ for the handles it produces over the course of the experiment:

$$\begin{aligned}
\mathbb{G}_1 \quad &\begin{cases} \langle \alpha \rangle \mapsto \tilde{\alpha} \\ \langle c_2^* \rangle \mapsto \tilde{s} \\ \langle \gamma_i \rangle \mapsto \tilde{\gamma}_i \end{cases} \quad \text{for } i \in [L] && \langle c_{3,i}^* \rangle \mapsto \begin{cases} \tilde{s}\tilde{\gamma}_i & i \in S^* \\ \tilde{r}_i & \text{otherwise} \end{cases} \quad \text{for } i \in [L], \\
\mathbb{G}_2 \quad &\begin{cases} \langle u_i \rangle \mapsto \tilde{u}_i & \text{for } i \in [K] \\ \langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi} & \text{for } \xi \in [q_{\text{KG}}], i \in S_\xi \end{cases} && \langle \delta_\xi \rangle \mapsto \tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \sum_{i \in S_\xi} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} \quad \text{for } \xi \in [q_{\text{KG}}], \\
\mathbb{G}_T \quad &\begin{cases} \langle c_1^* \rangle \mapsto \tilde{s} \tilde{\alpha} \sum_{i \in [K]} \text{id}^{*i-1} \tilde{u}_i + \tilde{m}_b^* \end{cases}
\end{aligned} \tag{A.1}$$

where $\tilde{m}_b^* = \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ is the challenger's formal polynomial associated with the challenge message handle $\langle m_b^* \rangle$. We also recall that ϕ_t maps each challenger variable to its corresponding sampled value (i.e., $\tilde{\alpha} \mapsto \alpha$, $\tilde{s} \mapsto s$, $\tilde{\gamma}_i \mapsto \gamma_i$, $\tilde{u}_i \mapsto u_i$, $\tilde{\beta}_{i,\xi} \mapsto \beta_{i,\xi}$, and $\tilde{r}_i \mapsto c_{3,i}^*$ for $i \notin S^*$), provided the value has been sampled by time t (otherwise, it is mapped to 0).

We now argue that $\text{Eval}_{\phi_t}(\tilde{h}) = x$ for every handle $\langle h \rangle$ with $\mathcal{T}(\langle h \rangle) = (x, v)$ and every time $t \geq t_h$, where t_h denotes the time at which $\langle h \rangle$ is created. For this, it suffices to show that $\text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$, as $\text{Eval}_{\phi_t}(\tilde{h}) = \text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$ for all $t \geq t_h$ because once an actual exponent is sampled, it is never reassigned. Our analysis breaks down into a case covering handles mapping to values directly sampled by the challenger (i.e., handles output by the challenger during setup, key-generation queries, and the challenge phase, excluding the message-dependent challenge-ciphertext component) which we call *directly generated handles*, and a case covering handles mapping to values computed from values stored in other handles (i.e., handles produced by generic group oracle queries as well as the message-dependent challenge-ciphertext component) which we call *derived handles*.

- **Case I: $\langle h \rangle$ is directly generated:** For this case, we can see from Eq. (A.1) that, by construction, each handle produced by the challenger during these phases maps directly to the polynomial that, when substituted with the

appropriate variables (as is done when executing $\text{Eval}_{\phi_{t_h}}(\tilde{h})$), exactly computes what the challenger computes to produce these exponents. We accordingly only need to argue that the appropriate variables have been sampled before time t_h , as they are otherwise substituted with 0.

For handles $\langle \alpha \rangle$, $\langle \gamma_i \rangle$, $\langle u_i \rangle$, $\langle \beta_{i,\xi} \rangle$, and $\langle c_2^* \rangle$, the challenger adds the sampled exponents α , γ_i , u_i , $\beta_{i,\xi}$, and s , to $\mathcal{T}$ directly before adding the challenger formal polynomials $\tilde{\alpha}$, $\tilde{\gamma}_i$, $\tilde{u}_i$, $\tilde{\beta}_{i,\xi}$, and $\tilde{s}$ to $\tilde{\mathcal{T}}$.

For a handle $\langle c_{3,i}^* \rangle$, recall that the challenger computes the exponent $c_{3,i}^* = s\gamma_i$ if $i \in S^*$ and samples $c_{3,i}^* \xleftarrow{R} \mathbb{Z}_p$ otherwise, and that this occurs in the challenge phase. In the former case, the entry is $\tilde{s}\tilde{\gamma}_i$ and both s and γ_i have been sampled by this point (in the challenge phase and setup phase respectively), while in the latter case, the entry is $\tilde{r}_i$ which is added to $\tilde{\mathcal{T}}$ directly after sampling $c_{3,i}^*$.

Finally, for a handle $\langle \delta_\xi \rangle$ which is sampled during a key-generation query, the challenger computes the exponent $\delta_\xi = \alpha h_\xi + \sum_{i \in S_\xi} \gamma_i \beta_{i,\xi}$ where $h_\xi = \sum_{i \in [K]} \text{id}_\xi^{i-1} u_i$, and the values α , $u_1, \dots, u_K$, and $\gamma_i, \beta_{i,\xi}$ for $i \in S_\xi$ were all sampled prior (in setup or earlier in the key-generation query in question).

- **Case II: $\langle h \rangle$ is derived:** We use induction on the order in which the handles were sampled.

The base case is the first handle created over the course of the experiment, which is a handle $\langle u_i \rangle$ that the challenger produces during setup. This is directly covered by Case I.

For the inductive step, we first consider each of the three generic group oracles separately. For the BGGM.Add and BGGM.Pair oracles, we write $\mathcal{T}(\langle h_1 \rangle) = (x_1, v_1)$ and $\mathcal{T}(\langle h_2 \rangle) = (x_2, v_2)$, and we note that the handles $\langle h_1 \rangle$ and $\langle h_2 \rangle$ must be sampled before $\langle h \rangle$ for the oracle to not abort. Thus, by the induction hypothesis, we have that $\text{Eval}_{\phi_{t_h}}(\tilde{h}_1) = x_1$ and $\text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_2$ at the time t_h .

- BGGM.Encode(x, v): The challenger adds the entry $\langle h \rangle \mapsto (x, v)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, thus $\text{Eval}_{\phi_{t_h}}(\tilde{h}) = x$.
- BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$): The challenger adds the entry $\langle h \rangle \mapsto (x_1 + x_2, v_1)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto \tilde{h}_1 + \tilde{h}_2$ to $\tilde{\mathcal{T}}$. Notably, every variable in $\tilde{h}_1 + \tilde{h}_2$ appears in $\tilde{h}_1$ or in $\tilde{h}_2$, and has thus already been sampled by time t_{h_1} or t_{h_2} where $t_{h_1}, t_{h_2} < t_h$. Furthermore, because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{h}_1 + \tilde{h}_2) = \text{Eval}_{\phi_{t_h}}(\tilde{h}_1) + \text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_1 + x_2.$$

- BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$): The challenger adds the entry $\langle h \rangle \mapsto (x_1 x_2, T)$ to $\mathcal{T}$ and the entry $\langle h \rangle \mapsto \tilde{h}_1 \cdot \tilde{h}_2$ to $\tilde{\mathcal{T}}$. Notably, every variable in $\tilde{h}_1 \cdot \tilde{h}_2$ appears in $\tilde{h}_1$ or in $\tilde{h}_2$, and has thus already been sampled by time t_{h_1} or t_{h_2} where $t_{h_1}, t_{h_2} < t_h$. Furthermore, because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}(\tilde{h}_1 \cdot \tilde{h}_2) = \text{Eval}_{\phi_{t_h}}(\tilde{h}_1) \cdot \text{Eval}_{\phi_{t_h}}(\tilde{h}_2) = x_1 x_2.$$

Finally, we analyze the handle $\langle c_1^* \rangle$, which the challenger produces during the challenge phase by executing BGGM.Encode($s\alpha h^* + m_b^*, T$) to add the entry $\langle c_1^* \rangle \mapsto (s\alpha h^* + m_b^*, T)$ to $\mathcal{T}$, where $\mathcal{T}(\langle m_b^* \rangle) = (m_b^*, T)$ is the entry for the challenge message handle. The challenger also adds $\langle c_1^* \rangle \mapsto \tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}^{*i-1} \tilde{u}_i + \tilde{m}_b^*$ to $\tilde{\mathcal{T}}$. As above, the values s , α , and $u_1, \dots, u_K$ were all sampled prior in the challenge phase or in setup. Furthermore, the challenge message handle $\langle m_b^* \rangle$ must have been sampled before the adversary $\mathcal{A}$ submits it and thus before $\langle c_1^* \rangle$ is sampled. By the induction hypothesis $\text{Eval}_{\phi_{t_h}}(\tilde{m}_b^*) = m_b^*$, and because $\text{Eval}_{\phi_{t_h}}$ is a ring homomorphism, we have that

$$\text{Eval}_{\phi_{t_h}}\left(\tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}^{*i-1} \tilde{u}_i + \tilde{m}_b^*\right) = s\alpha \sum_{i \in [K]} \text{id}^{*i-1} u_i + m_b^* = s\alpha h^* + m_b^*.$$

As we have shown above, the challenger's formal polynomial $\tilde{h}$ associated with a handle $\langle h \rangle$ always evaluates to the exponent x held in $\mathcal{T}(\langle h \rangle) = (x, v)$, and the lemma follows. Before continuing, we remark that we can also consider a mapping from the adversary's formal polynomials to challenger formal polynomials given by:

$$\begin{aligned} \mathbb{G}_1 \begin{cases} \hat{\alpha} \mapsto \tilde{\alpha} \\ \hat{c}_2^* \mapsto \tilde{s} \\ \hat{y}_i \mapsto \tilde{y}_i \quad \text{for } i \in [L] \end{cases} & \quad \hat{c}_{3,i}^* \mapsto \begin{cases} \tilde{s}\tilde{y}_i & i \in S^* \\ \tilde{r}_i & \text{otherwise} \end{cases} \quad \text{for } i \in [L], \\ \mathbb{G}_2 \begin{cases} \hat{u}_i \mapsto \tilde{u}_i & \text{for } i \in [K] \\ \hat{\beta}_{i,\xi} \mapsto \tilde{\beta}_{i,\xi} & \text{for } (i, \xi) \in [L] \times [q_{\text{KG}}] \end{cases} & \quad \hat{\delta}_\xi \mapsto \tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \quad \text{for } \xi \in [q_{\text{KG}}], \\ \mathbb{G}_T \begin{cases} \hat{c}_1^* \mapsto \tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}^{*i-1} \tilde{u}_i + \tilde{m}_b^*. \end{cases} & \end{aligned} \quad (\text{A.2})$$

where $\tilde{m}_b^* = \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ is the challenger's formal polynomial associated with the challenge message handle $\langle m_b^* \rangle$.

We can extend this to a ring homomorphism $\text{MapP}: \mathbb{Z}_p[\hat{\mathcal{V}}] \rightarrow \mathbb{Z}_p[\tilde{\mathcal{V}}]$ in the natural way, such that $\tilde{h} = \text{MapP}(\hat{h})$ for every handle $\langle h \rangle$, where $\hat{h} = \hat{\mathcal{T}}(\langle h \rangle)$ and $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ are the adversary's and the challenger's formal polynomials associated with $\langle h \rangle$. This holds by construction for the handles the challenger produces, as can be seen by inspection of Eq. (A.2), and it is preserved by the generic group oracles, as each of them updates $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ with the same ring operation (a constant polynomial for encode oracle queries, and the sum or the product of the entries of the queried handles for add and pair oracle queries), which MapP respects.

Note that because MapP is a ring homomorphism, it acts on the adversary's formal polynomial term by term (i.e., each term of $\hat{h}$ is mapped to the product of the challenger polynomials of the adversary variables it comprises, and $\tilde{h}$ is the sum of the resulting polynomials). We accordingly say that a term of $\hat{h}$ *induces* the monomials over $\tilde{\mathcal{V}}$ that appear in its image, and, when analyzing the subsequent hybrids, we will study $\tilde{h}$ by asking which terms of $\hat{h}$ can induce a given monomial. Notably, several terms of $\hat{h}$ may induce the same monomial, in which case their contributions to $\tilde{h}$ may cancel. $\square$

Lemma A.14. *There exists a negligible function $\text{negl}(\cdot)$ such that, for all $\lambda \in \mathbb{N}$,*

$$|\Pr[\text{Hyb}_3^{(b)}(\lambda)] - \Pr[\text{Hyb}_4^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. The hybrids are equivalent except for the event where the zero-test oracle is queried on a handle $\langle h \rangle$ associated with challenger formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ where $\text{Eval}_{\phi_t}(\tilde{h}) = 0$ (so $\text{Hyb}_3^{(b)}$ returns “zero”) but $\tilde{h} \neq 0$ (so $\text{Hyb}_4^{(b)}$ returns “non-zero”). Let bad denote this event. For a particular zero-test query on handle $\langle h' \rangle$ at time t' , we first observe that the associated polynomial $\tilde{h}'$ is at most degree 4. This is because the construction gives out elements as follows:

- In setup, key generation, and encryption, the adversary formal variables in $\hat{\mathcal{V}}_1$ (i.e., the left group $\mathbb{G}_1$) or $\hat{\mathcal{V}}_2$ (i.e., the right group $\mathbb{G}_2$) are sums of products of at most two challenger variables in $\tilde{\mathcal{V}}$, so $\mathcal{A}$ cannot pair to produce any polynomial of degree greater than 4.
- Only encryption outputs a component in the target group $\mathbb{G}_T$, which is a sum of products of three variables in $\tilde{\mathcal{V}}$, together with a challenge message polynomial produced by pairing other elements (which satisfy the case above).

As hybrids $\text{Hyb}_3^{(b)}$ and $\text{Hyb}_4^{(b)}$ are identical until bad, it suffices to bound $\Pr[\text{bad}]$ in either hybrid [BR06]. In $\text{Hyb}_4^{(b)}$, the zero-test oracle answers by checking whether $\tilde{h} \equiv 0$ and the exponents in $\mathcal{T}$ are never used, so $\mathcal{A}$'s view, and hence each queried polynomial $\tilde{h}'$, is independent of the sampled exponents. We may therefore apply the Schwartz-Zippel lemma (Lemma 2.3) to each zero-test query as though $\tilde{h}'$ were fixed in advance. We have that $\Pr[\text{Eval}_{\phi_{t'}}(\tilde{h}') = 0 \wedge \tilde{h}' \neq 0] \leq 4/p$. The lemma follows by applying the union bound and using the fact that $q_{\text{ZT}} = \text{poly}_3(\lambda)$ and $p \geq 2^\lambda$.

$$\Pr[\text{bad}] \leq q_{\text{ZT}} \cdot \Pr[\text{Eval}_{\phi_{t'}}(\tilde{h}') = 0 \wedge \tilde{h}' \neq 0] \leq \frac{4 \cdot q_{\text{ZT}}}{p} = \text{negl}(\lambda). \quad \square$$

Lemma A.15. For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_4^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,0,\text{int}}^{(b)}(\lambda)]$.

Proof. All that differs between the two hybrids is when the challenger samples the set S^* and performs the substitution (via the ring homomorphism Sub_{S^*}) on the entries of $\tilde{\mathcal{T}}$. Specifically, this step is performed at the start of the post-challenge query phase in hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$, while in hybrid $\text{Hyb}_4^{(b)}$ it is instead performed at the end of the challenge phase. That is, the challenger directly records the substituted entry $\langle c_{3,i}^* \rangle \mapsto \tilde{s}\tilde{y}_i$ for each $i \in S^*$ in hybrid $\text{Hyb}_4^{(b)}$ rather than the entry $\langle c_{3,i}^* \rangle \mapsto \tilde{r}_i$ as is done in hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$. We will argue that $\mathcal{A}$'s view is identical in both hybrids.

We first note that, starting in hybrid $\text{Hyb}_4^{(b)}$ (and for all remaining hybrids), the zero-test oracle answers a query on a handle $\langle h \rangle$ by testing whether $\tilde{h} \equiv 0$ (or not checking $\tilde{h}$ at all), and nothing derived from exponents from the table $\mathcal{T}$ is given to $\mathcal{A}$. It follows that the set S^* influences $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$.

Now, observe that $\mathcal{A}$'s view through the end of the challenge phase is identical in both hybrids, as the two hybrids differ only in the entries the challenger records in $\tilde{\mathcal{T}}$ for the handles $\langle c_{3,i}^* \rangle$. This is because $\mathcal{A}$ cannot make any generic group oracle queries between the end of the challenge phase and the substitution point at the start of the post-challenge query phase (notably, no handles are sampled and no zero-test queries are made). In particular, the handles present in $\tilde{\mathcal{T}}$ at the substitution point are exactly those present at the end of the challenge phase.

It remains to argue that the entries of $\tilde{\mathcal{T}}$ coincide in both hybrids from the substitution point onwards. Consider any handle $\langle h \rangle$ present in $\tilde{\mathcal{T}}$ and let $\tilde{h}'$ denote the entry it would have in the absence of any substitution (i.e., the entry obtained by recording $\langle c_{3,i}^* \rangle \mapsto \tilde{r}_i$ for every $i \in [L]$). We argue, by induction on the order in which the handles are created, that the entry $\langle h \rangle \mapsto \text{Sub}_{S^*}(\tilde{h}')$ is inserted in $\tilde{\mathcal{T}}$ in hybrids $\text{Hyb}_4^{(b)}$ and $\text{Hyb}_{5,0,\text{int}}^{(b)}$ and thus the behavior of the hybrids is equivalent.

For the base case, consider a handle $\langle h \rangle$ stored in $\tilde{\mathcal{T}}$ at the end of the challenge phase. We first analyze hybrid $\text{Hyb}_4^{(b)}$. If $\langle h \rangle = \langle c_{3,i}^* \rangle$ for some $i \in [L]$, then the hybrid records the entry $\langle h \rangle \mapsto \tilde{s}\tilde{y}_i$ if $i \in S^*$ and $\langle h \rangle \mapsto \tilde{r}_i$ otherwise, which in either case is exactly $\text{Sub}_{S^*}(\tilde{r}_i) = \text{Sub}_{S^*}(\tilde{h}')$ by the definition of Sub_{S^*} . Thus the mapping $\langle h \rangle \mapsto \text{Sub}_{S^*}(\tilde{h}')$ is added to $\tilde{\mathcal{T}}$. Otherwise, $\tilde{h}'$ does not contain any term involving any of the variables $\{\tilde{r}_i\}_{i \in [L]}$. In particular, by inspection of Eq. (A.1), the entries $\langle c_{3,i}^* \rangle \mapsto \tilde{r}_i$ are the only ones the challenger adds that induce such a variable and they are added at the very end of the challenge phase, while every adversary-added entry is added by a generic group oracle query and is either a constant polynomial or a sum/product of entries created earlier. Hence, $\text{Sub}_{S^*}(\tilde{h}') = \tilde{h}'$, and so the entry recorded is again $\langle h \rangle \mapsto \text{Sub}_{S^*}(\tilde{h}')$. Moving on to hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$, the base case holds by definition.

For the inductive step, consider a handle $\langle h \rangle$ created after the start of the post-challenge phase and suppose that every handle $\langle h_0 \rangle$ created before it has entry $\text{Sub}_{S^*}(\tilde{h}'_0)$. We go over each way $\langle h \rangle$ may have been produced below:

- **Post-challenge key-generation queries:** On the ξ^{th} key-generation query, the challenger adds the entries $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ for each $i \in S_\xi$ and $\langle \delta_\xi \rangle \mapsto \tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi}$ to $\tilde{\mathcal{T}}$. Neither entry contains any of the variables $\{\tilde{r}_i\}_{i \in [L]}$ and so both are unaffected by the Sub_{S^*} ring homomorphism.
- **Post-challenge generic group oracle queries:** The analysis of the generic group oracle queries is as follows:
 - **BGGM.Encode(x, v):** The challenger adds the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, which is again unaffected by the Sub_{S^*} ring homomorphism.
 - **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** Recall that the challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that Sub_{S^*} is a ring homomorphism, we have that
$$\tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{Sub}_{S^*}(\tilde{h}'_1) + \text{Sub}_{S^*}(\tilde{h}'_2) = \text{Sub}_{S^*}(\tilde{h}'_1 + \tilde{h}'_2) = \text{Sub}_{S^*}(\tilde{h}').$$
 - **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** The challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that Sub_{S^*} is a ring homomorphism, we have that
$$\tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{Sub}_{S^*}(\tilde{h}'_1) \cdot \text{Sub}_{S^*}(\tilde{h}'_2) = \text{Sub}_{S^*}(\tilde{h}'_1 \cdot \tilde{h}'_2) = \text{Sub}_{S^*}(\tilde{h}').$$

In each case, the entry $\langle h \rangle \mapsto \text{Sub}_{S^*}(\tilde{h}')$ is added to $\tilde{\mathcal{T}}$, and thus $\mathcal{A}$'s view is identical in the two hybrids. It follows that

$$\Pr[\text{Hyb}_4^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,0,\text{int}}^{(b)}(\lambda)]. \quad \square$$

Lemma A.16. For $L - L_{\text{ct}} = \omega(\log \lambda)$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ and all $\chi \in [q_{\text{ZT}}]$,

$$|\Pr[\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}(\lambda)] - \Pr[\text{Hyb}_{5,\chi}^{(b)}(\lambda)]| \leq \text{negl}(\lambda).$$

Proof. All that differs between the two hybrids is the behavior of the zero-test oracle on the χ^{th} post-challenge query the adversary $\mathcal{A}$ makes. Specifically, for said zero-test query on handle $\langle h \rangle$, the oracle checks if the challenger's formal polynomial $\tilde{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ is identically zero in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$, while it instead automatically returns “non-zero” when certain message- and challenge-set-dependent terms are present in the adversary's formal polynomial $\hat{h} = \hat{\mathcal{T}}(\langle h \rangle)$ in hybrid $\text{Hyb}_{5,\chi}^{(b)}$. We will argue that, except with negligible probability, the challenger's formal polynomial $\tilde{h}$ in this case would anyway not be identically zero and the oracle would output “non-zero” in any case. We let bad denote the event that $\tilde{h}$ (corresponding to an $\hat{h}$ that has the challenge message- or challenge-set-dependent terms) is identically zero.

Before continuing, we recall how the challenger's formal polynomials relate to the adversary's in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$. Let MapP_{int} denote the ring homomorphism defined as MapP , except that $\hat{c}_{3,i}^* \mapsto \tilde{r}_i$ for every $i \in [L]$ (matching the entries $\langle c_{3,i}^* \rangle \mapsto \tilde{r}_i$ recorded in the challenge phase of $\text{Hyb}_{5,0,\text{int}}^{(b)}$). By the same inductive argument as in Lemma A.13, $\tilde{h} = \text{MapP}_{\text{int}}(\hat{h})$ for every handle $\langle h \rangle$ up to the substitution point, which in this hybrid comes right after the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query. From that point onwards, by the inductive argument of Lemma A.15 (and since SubS_{S^*} is a ring homomorphism applied to every entry), $\tilde{h} = (\text{SubS}_{S^*} \circ \text{MapP}_{\text{int}})(\hat{h})$. Hence, for any χ^{th} post-challenge zero-test query, the challenger's formal polynomial in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$ can be derived from the adversary's formal polynomial via the following substitutions:

$$\begin{aligned} \mathbb{G}_1 \quad & \begin{cases} \hat{\alpha} \mapsto \tilde{\alpha} \\ \hat{c}_2^* \mapsto \tilde{s} \\ \hat{y}_i \mapsto \tilde{y}_i \quad \text{for } i \in [L] \end{cases} & \hat{c}_{3,i}^* \mapsto \begin{cases} \tilde{s}\tilde{y}_i & i \in S^* \wedge \chi' \geq \chi \\ \tilde{r}_i & \text{otherwise} \end{cases} \quad \text{for } i \in [L], \\ \mathbb{G}_2 \quad & \begin{cases} \hat{u}_i \mapsto \tilde{u}_i & \text{for } i \in [K] \\ \hat{\beta}_{i,\xi} \mapsto \tilde{\beta}_{i,\xi} & \text{for } (i,\xi) \in [L] \times [q_{\text{KG}}] \end{cases} & \hat{\delta}_\xi \mapsto \tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \quad \text{for } \xi \in [q_{\text{KG}}], \\ \mathbb{G}_T \quad & \begin{cases} \hat{c}_1^* \mapsto \tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}^{i-1} \tilde{u}_i + \tilde{m}_b^*. \end{cases} \end{aligned} \tag{A.3}$$

where $\tilde{m}_b^* = \tilde{\mathcal{T}}(\langle m_b^* \rangle)$ is the challenger's formal polynomial associated with the challenge message handle $\langle m_b^* \rangle$.

We are now ready to go over our main analysis of the χ^{th} zero-test query, which broadly breaks down into three cases. The first two analyze what happens when the adversary “combines” fewer than or at least K keys respectively to break the system, while the third covers remaining edge cases. In more detail (but still at a high level), for an adversary to distinguish the two hybrids, it must query an adversary formal polynomial $\hat{h}$ that maps to an identically zero challenger formal polynomial $\tilde{h}$ where $\hat{h}$ has a $\hat{c}_1^*$ term and/or a $\hat{c}_{3,i}^* \hat{v}$ term for some $\hat{v} \in \{1\} \cup \mathcal{V}_2$, per the conditions added in the subsequent hybrid. Below, we provide intuition for which terms in $\hat{h}$ are necessary to keep track of in our analysis (this applies to both the cases of combining fewer than and greater than or equal to K keys).

If $\hat{h}$ has a $\hat{c}_1^*$ term, then by construction this induces $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms in $\tilde{h}$, which the adversary can attempt to cancel out by including other terms in $\hat{h}$ which induce $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms in $\tilde{h}$. By inspection of Eq. (A.3) (and as we will justify in more detail later), $\tilde{s}\tilde{\alpha}\tilde{u}_i$ can only otherwise be obtained by pairing $\hat{c}_2^*$ with some $\hat{\delta}_\xi$ (i.e., $\hat{h}$ must include $\hat{c}_2^* \hat{\delta}_\xi$ terms for any cancellation to occur).

Note that the $\hat{c}_2^* \hat{\delta}_\xi$ terms in $\hat{h}$ also induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$, and these can only be cancelled by including more terms in $\hat{h}$ which induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$. As shown in Eq. (A.3) (and as we will again justify in more detail a bit later), the only other way to obtain $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ is to pair $\hat{c}_{3,i}^*$ with some $\hat{\beta}_{i,\xi}$ (i.e., $\hat{h}$ must include $\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}$ terms for any cancellation to occur).

For any terms of the form $\hat{c}_{3,i}^* \hat{v}$ for $\hat{v} \in \{1\} \cup \mathcal{V}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]}$ (i.e., the other challenge-set-dependent terms), a simple linear independence argument shows that cancellation does not occur if they are present.

To formalize the above intuition, consider the following events based on the adversary's formal polynomial $\hat{h}$:

$$\begin{aligned} E_1 &:= [\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0], \\ E_2 &:= [\text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0 \text{ for some } (i, j, \xi) \in [L]^2 \times [q_{\text{KG}}]], \\ E_3 &:= [\text{coeff}_{\hat{c}_{3,i}^* \hat{v}}(\hat{h}) \neq 0 \text{ for some } (i, \hat{v}) \in [L] \times (\{1\} \cup \hat{\mathcal{V}}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]})]. \end{aligned}$$

By construction, we have that E_1 is the message-dependent condition and $E_2 \vee E_3$ is the challenge-set-dependent condition, and thus the bad event can only occur in events E_1 , E_2 , or E_3 . Next, define $D = \{\xi \in [q_{\text{KG}}] : \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \neq 0\}$ and consider the following event that captures when the adversary “combines” fewer than K keys:

$$E_4 := [|D| < K].$$

With the above events, we can bound the probability of the bad event as

$$\begin{aligned} \Pr[\text{bad}] &= \Pr[\text{bad} \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge (E_1 \vee E_2 \vee E_3) \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &= \Pr[\text{bad} \wedge (E_1 \vee (\neg E_1 \wedge E_2) \vee E_3) \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge E_1 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4] + \Pr[\text{bad} \wedge E_3 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_4] \\ &\leq \Pr[\text{bad} \wedge E_1 \wedge E_4] + \Pr[\text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4] + \Pr[\text{bad} \wedge E_3] + \Pr[\text{bad} \wedge \neg E_4] \\ &= \Pr[\text{bad}_{\text{I-a}}] + \Pr[\text{bad}_{\text{I-b}}] + \Pr[\text{bad}_{\text{III}}] + \Pr[\text{bad}_{\text{II}}], \end{aligned}$$

where

$$\begin{aligned} \text{bad}_{\text{I-a}} &:= \text{bad} \wedge E_1 \wedge E_4 & \text{bad}_{\text{I-b}} &:= \text{bad} \wedge \neg E_1 \wedge E_2 \wedge E_4 \\ \text{bad}_{\text{II}} &:= \text{bad} \wedge \neg E_4 & \text{bad}_{\text{III}} &:= \text{bad} \wedge E_3. \end{aligned}$$

We now analyze the individual bad sub-cases below.

- **Case I: $\hat{h}$ combines fewer than K keys:** We begin by analyzing the two bad events in which $|D| < K$. As mentioned before, within this case we will analyze two sub-cases.

- **Case I-a: $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$:** At a high level, we analyze this case by arguing that a $\hat{c}_1^*$ term in $\hat{h}$ induces $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms in $\tilde{h}$, which can only be cancelled by $\hat{c}_2^* \hat{\delta}_\xi$ terms in $\hat{h}$ (as these are the only other terms that also induce $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms in $\tilde{h}$). If fewer than K different $\hat{c}_2^* \hat{\delta}_\xi$ terms are present, such a cancellation cannot occur.

We now identify which terms (that are not $\hat{c}_1^*$) in $\hat{h}$ can induce $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms in $\tilde{h}$. From Eq. (A.3), we can see that there are two types of terms in $\hat{h}$ that can induce the variable $\tilde{s}$ in $\tilde{h}$. These are $\hat{c}_2^*$ and $\hat{c}_{3,j}^*$ for a subset of $j \in [L]$. First observe that $\hat{c}_{3,j}^*$ induces $\tilde{s}\tilde{y}_j$ or $\tilde{r}_j$; in either case, this includes a variable that is not in $\tilde{s}\tilde{\alpha}\tilde{u}_i$, and so $\hat{c}_{3,j}^*$ cannot induce this (even if paired with other variables). Next, note that $\hat{c}_2^* \in \hat{\mathcal{V}}_1$ induces $\tilde{s}$ and thus must be paired with some $\hat{v} \in \hat{\mathcal{V}}_2$ that induces terms of the form $\tilde{\alpha}\tilde{u}_i$. From Eq. (A.3), we can see that $\hat{\delta}_\xi$ for $\xi \in [q_{\text{KG}}]$ is the only type of variable in $\hat{\mathcal{V}}_2$ that satisfies this.

Recalling that $D = \{\xi \in [q_{\text{KG}}] : \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \neq 0\}$, we can write the challenger's formal polynomial $\tilde{h}$ corresponding to the adversary's formal polynomial $\hat{h}$ on the χ^{th} query as

$$\tilde{h} = \tilde{\mu} + \text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \left(\tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}^{i-1} \tilde{u}_i \right) + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \left(\tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \tilde{s} \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) \quad (\text{A.4})$$

where $\tilde{\mu}$ is a polynomial over $\tilde{\mathcal{V}}$ such that $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\tilde{\mu}) = 0$ for all $i \in [K]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms). This holds because, as argued above, $\hat{c}_1^*$ and $\hat{c}_2^* \hat{\delta}_\xi$ are the only terms in $\hat{h}$ that induce $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms.

Note, in particular, that while $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \tilde{m}_b^*$ is absorbed into $\tilde{\mu}$, the above holds because $\tilde{m}_b^*$ contains no $\tilde{s}\tilde{\alpha}\tilde{u}_i$ monomial, due to the fact that $\langle m_b^* \rangle$ must have been created before the challenge phase and $\tilde{s}$ does not appear in any $\tilde{\mathcal{T}}$ entry before then. Rearranging Eq. (A.4), we have that

$$\tilde{h} = \tilde{\mu}' + \sum_{i \in [K]} \tilde{s}\tilde{\alpha}\tilde{u}_i \cdot \left(\text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \text{id}^{*i-1} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \text{id}_\xi^{i-1} \right), \quad (\text{A.5})$$

where $\tilde{\mu}' = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \sum_{i \in S_\xi} \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$. Intuitively, we have absorbed the $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms into $\tilde{\mu}'$ so that $\tilde{\mu}'$ contains all the terms that are not core to the analysis of the $|D| < K$ case. By construction, we have that $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\tilde{\mu}') = 0$ for each $i \in [K]$.

Let $\mathbf{V} = V(\text{id}^*, \text{id}_{\xi_1}, \dots, \text{id}_{\xi_{|D|}})$ be a square Vandermonde matrix of dimension $|D| + 1$ (see Section 2) over id^* and the elements of $D = \{\xi_1, \dots, \xi_{|D|}\}$. Because $|D| < K$ (or equivalently, $|D| + 1 \leq K$), we can rewrite Eq. (A.5) as $\tilde{h} = \tilde{\mu}'' + \mathbf{x}^\top \mathbf{V} \mathbf{y}$ for

$$\mathbf{x} = (\tilde{s}\tilde{\alpha}\tilde{u}_1, \dots, \tilde{s}\tilde{\alpha}\tilde{u}_{|D|+1}) \quad \text{and} \quad \mathbf{y} = (\text{coeff}_{\hat{c}_1^*}(\hat{h}), \text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi_1}}(\hat{h}), \dots, \text{coeff}_{\hat{c}_2^* \hat{\delta}_{\xi_{|D|}}}(\hat{h})),$$

and where $\tilde{\mu}'' = \tilde{\mu}' + \sum_{i \in [|D|+2, K]} \tilde{s}\tilde{\alpha}\tilde{u}_i \cdot (\text{coeff}_{\hat{c}_1^*}(\hat{h}) \cdot \text{id}^{*i-1} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \text{id}_\xi^{i-1})$. Intuitively, we have absorbed the $\tilde{s}\tilde{\alpha}\tilde{u}_{|D|+2}, \dots, \tilde{s}\tilde{\alpha}\tilde{u}_K$ terms into $\tilde{\mu}''$, and so $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\tilde{\mu}'') = 0$ for each $i \in [|D| + 1]$.

Observe that if $\mathbf{V} \mathbf{y} \neq \mathbf{0}$, then $\mathbf{x}^\top \mathbf{V} \mathbf{y} \neq 0$. This is because we can express it as a sum $\mathbf{x}^\top \mathbf{V} \mathbf{y} = \sum_{i \in [|D|+1]} z_i \cdot \tilde{s}\tilde{\alpha}\tilde{u}_i$ where $\mathbf{V} \mathbf{y} = (z_1, \dots, z_{|D|+1})$, which is identically zero if and only if $z_i = 0$ for all $i \in [|D| + 1]$. Hence, in this case, $\tilde{h} = \tilde{\mu}'' + \mathbf{x}^\top \mathbf{V} \mathbf{y} \neq 0$, because $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\mathbf{x}^\top \mathbf{V} \mathbf{y}) \neq 0$ for some $i \in [|D| + 1]$ while $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\tilde{\mu}'') = 0$ for all $i \in [|D| + 1]$ as shown above (i.e., $\mathbf{x}^\top \mathbf{V} \mathbf{y}$ consists solely of $\tilde{s}\tilde{\alpha}\tilde{u}_1, \dots, \tilde{s}\tilde{\alpha}\tilde{u}_{|D|+1}$ terms and $\tilde{\mu}''$ does not contain any, so there are no terms in $\tilde{\mu}''$ that can cancel out terms in $\mathbf{x}^\top \mathbf{V} \mathbf{y}$).

Now, suppose for the sake of contradiction that $\mathbf{V} \mathbf{y} = \mathbf{0}$. Because $\mathbf{V}$ is a square Vandermonde matrix and each $\text{id}^*, \text{id}_{\xi_1}, \dots, \text{id}_{\xi_{|D|}} \in \{0, 1\}^\lambda$ are pair-wise distinct (as the adversary is admissible and we are working in the unique-identity setting), and $p > 2^\lambda$, then $\det(\mathbf{V}) \neq 0$ (i.e., $\mathbf{V}$ is invertible). It follows that $\mathbf{0} = \mathbf{V}^{-1} \mathbf{V} \mathbf{y} = \mathbf{y}$ (i.e., every coefficient in $\mathbf{y}$ is 0), but this contradicts the assumption for this case that $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$. It follows that $\tilde{h} \neq 0$ and hence the zero-test oracle always returns “non-zero” in this case. Thus,

$$\Pr[\text{bad}_{\Gamma\text{-a}}] = 0.$$

- **Case I-b: $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = 0$ and $\text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0$ for some $(i, j, \xi) \in [L]^2 \times [q_{\text{KG}}]$:** At a very high level, we analyze this case by arguing that a $\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}$ term in $\hat{h}$ induces $\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$. The former can only be cancelled by including $\hat{c}_2^* \hat{\delta}_\xi$ terms in $\hat{h}$ (as these are the only other terms that also induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}$). If fewer than K different $\hat{c}_2^* \hat{\delta}_\xi$ terms are present, such a cancellation cannot occur per a Vandermonde matrix argument (over $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms) similar to the previous case. For the latter, the lack of cancellation is straightforward but tedious.

We now identify which terms in $\hat{h}$ can induce terms of the form $\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ in $\tilde{h}$. By inspecting Eq. (A.3), we can see that there are two types of terms in $\hat{h}$ that can induce the formal variable $\tilde{\beta}_{j,\xi}$ in $\tilde{h}$. These are $\hat{\beta}_{j,\xi}$ and $\hat{\delta}_\xi$ which are both in $\hat{\mathcal{V}}_2$. For the former, it must be paired with some $\hat{v} \in \hat{\mathcal{V}}_1$ that induces terms of the form $\tilde{s}\tilde{y}_i$ or $\tilde{r}_i$. From Eq. (A.3), we can see that $\hat{c}_{3,i}^*$ is the only type of variable in $\hat{\mathcal{V}}_1$ that satisfies this. For the latter, it must be paired with some $\hat{v} \in \hat{\mathcal{V}}_1$ that induces a solo $\tilde{s}$ variable (i.e., without any additional variables multiplied), and from Eq. (A.3) we can see only $\hat{c}_2^*$ satisfies this.

Let $C = \{(i, j, \xi) \in [L]^2 \times [q_{\text{KG}}] : \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0\}$ be the set of indices for $\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}$ and define the following sets:

$$\begin{aligned} C' &= \{(i, \xi) : (i, j, \xi) \in C \wedge i = j\}, & C'' &= \{(i, j, \xi) \in C : i \neq j\}, \\ C'_{S^*} &= \{(i, \xi) \in C' : i \in S^*\}, & C''_{S^*} &= \{(i, j, \xi) \in C'' : i \in S^*\}. \end{aligned}$$

Intuitively, C' and C'' can be thought of as partitioning C into the “diagonal” and “off-diagonal” $\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}$. Furthermore, C'_{S^*} and C''_{S^*} are the subsets that induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$ (because $i \in S^*$) as opposed to $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms in $\tilde{h}$.

Recalling that $D = \{\xi \in [q_{KG}] : \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \neq 0\}$, we can write the challenger’s formal polynomial $\tilde{h}$ corresponding to the adversary’s formal polynomial $\hat{h}$ on the χ^{th} query as

$$\begin{aligned} \tilde{h} = & \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \left(\tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \tilde{s} \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) \\ & + \sum_{(i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot (\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}) + \sum_{(i,\xi) \in C' \setminus C'_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot (\tilde{r}_i \tilde{\beta}_{i,\xi}) \\ & + \sum_{(i,j,\xi) \in C''_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot (\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}) + \sum_{(i,j,\xi) \in C'' \setminus C''_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot (\tilde{r}_i \tilde{\beta}_{j,\xi}) \end{aligned} \quad (\text{A.6})$$

where $\tilde{\mu}$ is a polynomial over $\mathcal{V}$ such that $\text{coeff}_{\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = \text{coeff}_{\tilde{r}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = 0$ for all $(i, j, \xi) \in [L]^2 \times [q_{KG}]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms). This holds because, as argued above, $\hat{c}_2^* \hat{\delta}_\xi$ and $\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}$ are the only terms in $\hat{h}$ that can induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms. Furthermore, $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\tilde{\mu}) = 0$ for all $i \in [K]$ (i.e., $\tilde{\mu}$ also does not contain any $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms). This is because, as shown in Case I-a, $\hat{c}_1^*$ and $\hat{c}_2^* \hat{\delta}_\xi$ are the only terms in $\hat{h}$ that induce $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms. The former has $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = 0$ by the definition of this case, and every contribution of the latter already appears explicitly in the first sum of Eq. (A.6).

By inspection of Eq. (A.6) and using the above fact on $\tilde{\mu}$, we can see that

$$\begin{aligned} \text{coeff}_{\tilde{r}_i \tilde{\beta}_{i,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) && \text{for } (i, \xi) \in C' \setminus C'_{S^*}, \\ \text{coeff}_{\tilde{s}\tilde{y}_i \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C''_{S^*}, \\ \text{coeff}_{\tilde{r}_i \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C'' \setminus C''_{S^*}, \end{aligned}$$

(i.e., the only cancellation that can occur is on $\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}$ terms in $\hat{h}$ that induce $\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$). Thus, if C'_{S^*} , $C'' \setminus C''_{S^*}$, or $C' \setminus C'_{S^*}$ are non-empty, then $\tilde{h} \neq 0$.

Now suppose that C'_{S^*} , $C'' \setminus C''_{S^*}$, and $C' \setminus C'_{S^*}$ are all empty (note that this implies that C'_{S^*} must be non-empty, as we assume $\text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0$ for some $(i, j, \xi) \in [L]^2 \times [q_{KG}]$) in this case. We can simplify Eq. (A.6) to

$$\tilde{h} = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \left(\tilde{s}\tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \tilde{s} \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi} \right) + \sum_{(i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot (\tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi}). \quad (\text{A.7})$$

Rearranging Eq. (A.7) further, we have that

$$\begin{aligned} \tilde{h} = & \tilde{\mu} + \sum_{i \in [K]} \tilde{s}\tilde{\alpha}\tilde{u}_i \cdot \left(\sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \text{id}_\xi^{i-1} \right) + \sum_{(i,\xi) \in U \cap C'_{S^*}} \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \right) \\ & + \sum_{(i,\xi) \in U \setminus C'_{S^*}} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi} + \sum_{(i,\xi) \in C'_{S^*} \setminus U} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \tilde{s}\tilde{y}_i \tilde{\beta}_{i,\xi} \end{aligned} \quad (\text{A.8})$$

where $U = \bigcup_{\xi \in D} \{(i, \xi) : i \in S_\xi\}$ is the set of indices (i, ξ) such that $\xi \in D$ and $i \in S_\xi$ (i.e., the indices that appear in the left-hand sum in Eq. (A.7)).

For the remainder, we will argue that if D is empty, then Eq. (A.8) shows that $\tilde{h}$ is not identically zero; while, on the other hand, if D is non-empty, then we can rewrite Eq. (A.8) in terms of a Vandermonde matrix as was done in Case I-a, which we will then use to show that $\tilde{h}$ is again not identically zero.

Now suppose that D is empty. By inspection of Eq. (A.8) and using the fact C'_{S^*} is non-empty (as argued above), there exists some $(i, \xi) \in C'_{S^*}$ such that

$$\text{coeff}_{\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}}(\tilde{h}) = \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{3,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) = \text{coeff}_{\hat{c}_{3,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \neq 0,$$

and so $\tilde{h} \neq 0$ in this case.

Now suppose D is non-empty. First, we rewrite Eq. (A.8) as

$$\tilde{h} = \tilde{\mu}' + \sum_{i \in [K]} \tilde{s}\tilde{\alpha}\tilde{u}_i \cdot \left(\sum_{\xi \in D} \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \cdot \text{id}_\xi^{i-1} \right), \quad (\text{A.9})$$

where

$$\begin{aligned} \tilde{\mu}' &= \tilde{\mu} + \sum_{(i,\xi) \in U \cap C'_{S^*}} \tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{3,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \right) \\ &+ \sum_{(i,\xi) \in U \setminus C'_{S^*}} \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi} + \sum_{(i,\xi) \in C'_{S^*} \setminus U} \text{coeff}_{\hat{c}_{3,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \cdot \tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}. \end{aligned}$$

Notably, because $\tilde{\mu}$ contains no $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms and the three absorbed sums consist solely of $\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}$ terms, we have that $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\tilde{\mu}') = 0$ for all $i \in [K]$.

Now since D is non-empty, there exists some $\xi' \in D$ such that $\text{coeff}_{\hat{c}_2^*\hat{\delta}_{\xi'}}(\hat{h}) \neq 0$. Label the elements of $D = \{\xi_1, \dots, \xi_{|D|}\}$ where $\xi_1 = \xi'$ and let $\mathbf{V} = V(\text{id}_{\xi_1}, \dots, \text{id}_{\xi_{|D|}})$ be a square Vandermonde matrix over the elements of D . Because $|D| < K$, we can rewrite Eq. (A.8) as $\tilde{h} = \tilde{\mu}'' + \mathbf{x}^T \mathbf{V} \mathbf{y}$ for

$$\mathbf{x} = (\tilde{s}\tilde{\alpha}\tilde{u}_1, \dots, \tilde{s}\tilde{\alpha}\tilde{u}_{|D|}) \quad \text{and} \quad \mathbf{y} = (\text{coeff}_{\hat{c}_2^*\hat{\delta}_{\xi_1}}(\hat{h}), \dots, \text{coeff}_{\hat{c}_2^*\hat{\delta}_{\xi_{|D|}}}(\hat{h})),$$

and where $\tilde{\mu}'' = \tilde{\mu}' + \sum_{i=|D|+1}^K \tilde{s}\tilde{\alpha}\tilde{u}_i \cdot (\sum_{\xi \in D} \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \cdot \text{id}_\xi^{i-1})$. Intuitively, we have absorbed the $\tilde{s}\tilde{\alpha}\tilde{u}_{|D|+1}, \dots, \tilde{s}\tilde{\alpha}\tilde{u}_K$ terms into $\tilde{\mu}''$, and so $\text{coeff}_{\tilde{s}\tilde{\alpha}\tilde{u}_i}(\tilde{\mu}'') = 0$ for each $i \in [|D|]$.

Following the analysis in Case I-a, recall that if $\mathbf{V} \mathbf{y} \neq \mathbf{0}$, then $\mathbf{x}^T \mathbf{V} \mathbf{y} \neq 0$ and the challenger's formal polynomial $\tilde{h} = \tilde{\mu}'' + \mathbf{x}^T \mathbf{V} \mathbf{y} \neq 0$. Now, suppose for the sake of contradiction that $\mathbf{V} \mathbf{y} = \mathbf{0}$. Because $\mathbf{V}$ is a square Vandermonde matrix and each $\text{id}_{\xi_1}, \dots, \text{id}_{\xi_{|D|}}$ are pair-wise distinct (as we are working in the unique-identity setting) then $\det(\mathbf{V}) \neq 0$ and $\mathbf{0} = \mathbf{V}^{-1} \mathbf{V} \mathbf{y} = \mathbf{y}$ (i.e., every coefficient in $\mathbf{y}$ is 0). This contradicts the proposition that $\text{coeff}_{\hat{c}_2^*\hat{\delta}_{\xi_1}}(\hat{h}) = \text{coeff}_{\hat{c}_2^*\hat{\delta}_{\xi'}}(\hat{h}) \neq 0$. It again follows that $\tilde{h} \neq 0$ in this case.

Wrapping up, we have shown that if $\text{coeff}_{\hat{c}_{3,i}^*\hat{\beta}_{j,\xi}}(\hat{h}) \neq 0$ for some $(i, j, \xi) \in [L]^2 \times [q_{KG}]$, $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = 0$, and $|D| < K$, then there is no condition under which $\tilde{h}$ is identically zero. Thus,

$$\Pr[\text{bad}_{\text{I-b}}] = 0.$$

- **Case II: $\hat{h}$ combines K or more keys:** We now analyze the bad event in which $|D| \geq K$. As argued for Case I-b, including $\hat{c}_2^*\hat{\delta}_\xi$ terms in $\hat{h}$ induces $\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$, which must be cancelled by also including $\hat{c}_{3,i}^*\hat{\beta}_{i,\xi}$ terms in $\hat{h}$ (as these are the only other terms which induce terms of the form $\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}$). At a high level, we will show that including K different $\hat{c}_2^*\hat{\delta}_\xi$ terms in $\hat{h}$ will induce $\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}$ terms in $\tilde{h}$ for a sufficient number of $i \in [L]$ such that, with

high probability, there is some i' induced such that $i' \notin S^*$. For i' , the $\hat{c}_{3,i'}^* \hat{\beta}_{i',\xi}$ term needed in $\hat{h}$ induces a $\tilde{r}_{i'} \tilde{\beta}_{i',\xi}$ term as opposed to $\tilde{s} \tilde{\gamma}_{i'} \tilde{\beta}_{i',\xi}$ term (because $i' \notin S^*$), and thus cancellation cannot occur and $\tilde{h}$ is not identically zero.

Because the $\tilde{s} \tilde{\gamma}_i \tilde{\beta}_{i,\xi}$ terms we analyze are identical to those used in Case I-b, we directly proceed with the cancellation argument. Let $C = \{(i, j, \xi) \in [L]^2 \times [q_{KG}] : \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \neq 0\}$ be the set of indices for $\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}$ terms in $\hat{h}$ and define the following sets:

$$\begin{aligned} C' &= \{(i, \xi) : (i, j, \xi) \in C \wedge i = j\}, & C'' &= \{(i, j, \xi) \in C : i \neq j\}, \\ C'_{S^*} &= \{(i, \xi) \in C' : i \in S^*\}, & C''_{S^*} &= \{(i, j, \xi) \in C'' : i \in S^*\}. \end{aligned}$$

As in Case I-b, the challenger's formal polynomial can be written as

$$\begin{aligned} \tilde{h} &= \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \left(\tilde{s} \tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \tilde{s} \sum_{i \in S_\xi} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} \right) \\ &+ \sum_{(i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \left(\tilde{s} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} \right) + \sum_{(i,\xi) \in C' \setminus C'_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \left(\tilde{r}_i \tilde{\beta}_{i,\xi} \right) \\ &+ \sum_{(i,j,\xi) \in C''_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot \left(\tilde{s} \tilde{\gamma}_i \tilde{\beta}_{j,\xi} \right) + \sum_{(i,j,\xi) \in C'' \setminus C''_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) \cdot \left(\tilde{r}_i \tilde{\beta}_{j,\xi} \right) \end{aligned} \quad (\text{A.10})$$

where $\tilde{\mu}$ is a polynomial over $\mathcal{V}$ such that $\text{coeff}_{\tilde{s} \tilde{\gamma}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = \text{coeff}_{\tilde{r}_i \tilde{\beta}_{j,\xi}}(\tilde{\mu}) = 0$ for all $(i, j, \xi) \in [L]^2 \times [q_{KG}]$ (i.e., $\tilde{\mu}$ does not contain any $\tilde{s} \tilde{\gamma}_i \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{\beta}_{j,\xi}$ terms).

By inspection of Eq. (A.10) and using the above fact on $\tilde{\mu}$, we have

$$\begin{aligned} \text{coeff}_{\tilde{r}_i \tilde{\beta}_{i,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) && \text{for } (i, \xi) \in C' \setminus C'_{S^*}, \\ \text{coeff}_{\tilde{s} \tilde{\gamma}_i \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C''_{S^*}, \\ \text{coeff}_{\tilde{r}_i \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{j,\xi}}(\hat{h}) && \text{for } (i, j, \xi) \in C'' \setminus C''_{S^*}. \end{aligned}$$

Thus, if C''_{S^*} , $C'' \setminus C''_{S^*}$, or $C' \setminus C'_{S^*}$ are non-empty, then $\tilde{h} \neq 0$.

Now suppose that C''_{S^*} , $C'' \setminus C''_{S^*}$, and $C' \setminus C'_{S^*}$ are empty. We can simplify Eq. (A.10) to

$$\tilde{h} = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \left(\tilde{s} \tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \tilde{s} \sum_{i \in S_\xi} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} \right) + \sum_{(i,\xi) \in C'_{S^*}} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \left(\tilde{s} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} \right), \quad (\text{A.11})$$

Rearranging Eq. (A.11) further, we have that

$$\begin{aligned} \tilde{h} &= \tilde{\mu}' + \sum_{(i,\xi) \in U \cap C'_{S^*}} \tilde{s} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} \cdot \left(\text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) + \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \right) \\ &+ \sum_{(i,\xi) \in U \setminus C'_{S^*}} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \tilde{s} \tilde{\gamma}_i \tilde{\beta}_{i,\xi} + \sum_{(i,\xi) \in C'_{S^*} \setminus U} \text{coeff}_{\hat{c}_{3,i}^* \hat{\beta}_{i,\xi}}(\hat{h}) \cdot \tilde{s} \tilde{\gamma}_i \tilde{\beta}_{i,\xi}, \end{aligned} \quad (\text{A.12})$$

where $U = \bigcup_{\xi \in D} \{(i, \xi) : i \in S_\xi\}$ is the set of indices (i, ξ) such that $\xi \in D$ and $i \in S_\xi$ (i.e., the indices that appear in the top sum in Eq. (A.11)), and where

$$\tilde{\mu}' = \tilde{\mu} + \sum_{\xi \in D} \text{coeff}_{\hat{c}_2^* \hat{\delta}_\xi}(\hat{h}) \cdot \left(\sum_{i \in [K]} \tilde{s} \tilde{\alpha} \tilde{u}_i \cdot \text{id}_\xi^{i-1} \right).$$

Notably, we have absorbed the $\tilde{s}\tilde{\alpha}\tilde{u}_i$ terms into $\tilde{\mu}'$ so that $\tilde{\mu}'$ contains all the terms that are not core to the analysis of the $|D| \geq K$ case. By construction, we have that $\text{coeff}_{\tilde{s}\tilde{y}_i\tilde{\beta}_{j,\xi}}(\tilde{\mu}') = 0$ for all $(i, j, \xi) \in [L]^2 \times [q_{\text{KG}}]$.

Intuitively, we can see from Eq. (A.12) that the only cancellations can occur in $U \cap C'_{S^*}$. That is, the sets $U \setminus C'_{S^*}$ and $C'_{S^*} \setminus U$ must be empty (which we will formally argue below) and thus $C'_{S^*} = U$ for the bad event to occur. We conclude by arguing that the adversary has a negligible probability of producing an $\hat{h}$ such that $C'_{S^*} = U$.

Proceeding with the argument outlined above, if $U \neq C'_{S^*}$ then there exists some $(i, \xi) \in (U \setminus C'_{S^*}) \cup (C'_{S^*} \setminus U)$. It follows that $\text{coeff}_{\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}}(\hat{h}) = \text{coeff}_{\hat{c}_2^*\hat{\delta}_\xi}(\hat{h}) \neq 0$ or $\text{coeff}_{\tilde{s}\tilde{y}_i\tilde{\beta}_{i,\xi}}(\hat{h}) = \text{coeff}_{\hat{c}_{3,i}^*\hat{\beta}_{i,\xi}}(\hat{h}) \neq 0$, so $\hat{h} \neq 0$ in this case.

It remains to bound the probability that $C'_{S^*} = U$, which itself is bounded by the probability that $U \subseteq C'_{S^*}$. First we recall that $C'_{S^*} = \{(i, \xi) \in C' : i \in S^*\}$ and $U = \bigcup_{\xi \in D} \{(i, \xi) : i \in S_\xi\}$. Per the change added in hybrid $\text{Hyb}_{5,\chi-1,\text{int}}^{(b)}$, the set S^* is sampled after the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query, and so no zero-test query is answered between the sampling of S^* and the current query. Because S^* influences $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$, the adversary has not received any information on S^* , so we have that

$$\Pr[\text{bad}_{\Pi}] \leq \Pr[U \subseteq C'_{S^*}] \leq \Pr\left[\bigcup_{\xi \in D} S_\xi \subseteq S^*\right] \leq \left(\frac{L_{\text{ct}}}{L}\right)^{\frac{L}{\kappa}},$$

where the final inequality follows from the halting condition added in hybrid $\text{Hyb}_1^{(b)}$ which asks that there is no collection of sets $S_{\xi_1}, \dots, S_{\xi_K}$ with union size less than or equal to L/κ . Since $|D| \geq K$ in this case, this condition applies to any K elements of D and yields $|\bigcup_{\xi \in D} S_\xi| > L/\kappa$.

Finally, recall that $L - L_{\text{ct}} = \omega(\log \lambda)$ and $\kappa \geq 2$. Consider any $c_1 > 0$. Because $L - L_{\text{ct}} = \omega(\log \lambda)$, there exists some $d_1 \in \mathbb{N}$ such that $L - L_{\text{ct}} \geq c_1 \log \lambda$ for all $\lambda \geq d_1$. Let $c = c_1 \log e / \kappa$. We have that,

$$\begin{aligned} \Pr[\text{bad}_{\Pi}] &\leq \left(1 - \frac{L - L_{\text{ct}}}{L}\right)^{\frac{L}{\kappa}} \\ &\leq \exp\left(-\frac{L - L_{\text{ct}}}{\kappa}\right) \\ &\leq \exp\left(-\frac{c_1 \log \lambda}{\kappa}\right) && \text{for } \lambda \geq d_1 \\ &= \lambda^{-c_1 \log e / \kappa} && \text{for } \lambda \geq d_1 \\ &= \lambda^{-c} && \text{for } \lambda \geq d_1. \end{aligned}$$

Note that since $c_1 > 0$ was arbitrary, $c = c_1 \log e / \kappa$ can take any positive value, and so there exists a negligible function $\text{negl}(\cdot)$ such that

$$\Pr[\text{bad}_{\Pi}] = \text{negl}(\lambda).$$

- **Case III: $\text{coeff}_{\hat{c}_{3,i}^*\hat{\delta}}(\hat{h}) \neq 0$ for some $(i, \hat{\delta}) \in [L] \times (\{1\} \cup \mathcal{V}_2 \setminus \{\hat{\beta}_{j,\xi}\}_{(j,\xi) \in [L] \times [q_{\text{KG}}]})$:** This final case considers the remaining challenge-set-dependent terms not covered in the previous cases.

Explicitly, this case occurs if, for $i \in [L]$, $\text{coeff}_{\hat{c}_{3,i}^*}(\hat{h}) \neq 0$, $\text{coeff}_{\hat{c}_{3,i}^*\hat{u}_j}(\hat{h}) \neq 0$ for some $j \in [K]$, or $\text{coeff}_{\hat{c}_{3,i}^*\hat{\delta}_\xi}(\hat{h}) \neq 0$ for some $\xi \in [q_{\text{KG}}]$. We consider these as sub-cases and analyze them below:

- **Case III-a: $\text{coeff}_{\hat{c}_{3,i}^*}(\hat{h}) \neq 0$ for some $i \in [L]$:** A $\hat{c}_{3,i}^*$ term in $\hat{h}$ induces an $\tilde{s}\tilde{y}_i$ or $\tilde{r}_i$ term in $\tilde{h}$. We argue that no other terms induce $\tilde{s}\tilde{y}_i$ or $\tilde{r}_i$ terms in $\tilde{h}$, and thus no cancellation occurs in this case. From Eq. (A.3), we have that no other term in $\hat{h}$ induces a term of the form $\tilde{s}\tilde{y}_i$ and thus the only remaining option to induce it is to pair an element $\hat{v}_1 \in \mathcal{V}_1$ and $\hat{v}_2 \in \mathcal{V}_2$. If $\hat{v}_1$ is $\hat{c}_2^*$, then this induces an $\tilde{s}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \mathcal{V}_2$ contains a (solo) $\tilde{y}_i$ term. If $\hat{v}_1$ is $\hat{y}_i$, then this induces a $\tilde{y}_i$ term in $\tilde{h}$, however no $\hat{v}_2 \in \mathcal{V}_2$ contains

an $\tilde{s}$ term. Finally, we have that no other term in $\hat{h}$ induces a (solo) $\tilde{r}_i$. In particular, a $\hat{c}_1^*$ term induces neither an $\tilde{s}\tilde{y}_i$ nor an $\tilde{r}_i$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{s}$ nor any $\tilde{r}_i$ appears in a $\tilde{\mathcal{T}}$ entry before then.

- **Case III-b: $\text{coeff}_{\hat{c}_{3,i}^* \hat{u}_j}(\hat{h}) \neq 0$ for some $(i, j) \in [L] \times [K]$:** A $\hat{c}_{3,i}^* \hat{u}_j$ term in $\hat{h}$ induces an $\tilde{s}\tilde{y}_i \tilde{u}_j$ or $\tilde{r}_i \tilde{u}_j$ term in $\tilde{h}$. We argue that no other terms induce these terms in $\tilde{h}$. From Eq. (A.3), we can see that the only remaining option to induce a $\tilde{s}\tilde{y}_i \tilde{u}_j$ or $\tilde{r}_i \tilde{u}_j$ term, other than by pairing $\hat{c}_{3,i}^*$ with $\hat{u}_j$ itself, is to pair some other element $\hat{v}_1 \in \hat{\mathcal{V}}_1$ with $\hat{v}_2 \in \hat{\mathcal{V}}_2$. If $\hat{v}_1$ is $\hat{c}_2^*$, then this induces an $\tilde{s}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains a (solo) $\tilde{y}_i \tilde{u}_j$ term. If $\hat{v}_1$ is $\hat{y}_i$, then this induces a $\tilde{y}_i$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains an $\tilde{s}\tilde{u}_j$ term. If $\hat{v}_1$ is $\hat{c}_{3,i'}^*$ for some $i' \neq i$, then this induces an $\tilde{s}\tilde{y}_{i'}$ or $\tilde{r}_{i'}$ term in $\tilde{h}$, which (paired with any $\hat{v}_2 \in \hat{\mathcal{V}}_2$) cannot equal $\tilde{s}\tilde{y}_i \tilde{u}_j$ or $\tilde{r}_i \tilde{u}_j$ since $i' \neq i$. Finally, we have that $\hat{u}_j$ is the only element of $\hat{\mathcal{V}}_2$ which induces a (solo) $\tilde{u}_j$ term, so no other pairing can induce the required $\tilde{u}_j$ factor either. In particular, like Case III-a, a $\hat{c}_1^*$ term induces neither an $\tilde{s}\tilde{y}_i \tilde{u}_j$ nor an $\tilde{r}_i \tilde{u}_j$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{s}$ nor any $\tilde{r}_i$ appears in a $\tilde{\mathcal{T}}$ entry before then.
- **Case III-c: $\text{coeff}_{\hat{c}_{3,i}^* \hat{\delta}_\xi}(\hat{h}) \neq 0$ for some $(i, \xi) \in [L] \times [q_{\text{KG}}]$:** For $j \in S_\xi$, a $\hat{c}_{3,i}^* \hat{\delta}_\xi$ term in $\hat{h}$ induces an $\tilde{s}\tilde{y}_i \tilde{y}_j \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{y}_j \tilde{\beta}_{j,\xi}$ term in $\tilde{h}$. We argue that no other terms induce these terms in $\tilde{h}$. From Eq. (A.3), we can see that the only remaining option to induce a $\tilde{s}\tilde{y}_i \tilde{y}_j \tilde{\beta}_{j,\xi}$ or $\tilde{r}_i \tilde{y}_j \tilde{\beta}_{j,\xi}$ term, other than by pairing $\hat{c}_{3,i}^*$ with $\hat{\delta}_\xi$ itself, is to pair some other element $\hat{v}_1 \in \hat{\mathcal{V}}_1$ with $\hat{v}_2 \in \hat{\mathcal{V}}_2$. If $\hat{v}_1$ is $\hat{c}_2^*$, then this induces an $\tilde{s}$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains a (solo) $\tilde{y}_i \tilde{y}_j \tilde{\beta}_{j,\xi}$ term. If $\hat{v}_1$ is $\hat{y}_i$ or $\hat{y}_j$, then this induces a $\tilde{y}_i$ or $\tilde{y}_j$ term in $\tilde{h}$, however no $\hat{v}_2 \in \hat{\mathcal{V}}_2$ contains an $\tilde{s}$ or $\tilde{r}_i$ term. If $\hat{v}_1$ is $\hat{c}_{3,i'}^*$ for some $i' \neq i$, then it induces an $\tilde{s}\tilde{y}_{i'}$ or $\tilde{r}_{i'}$ term in $\tilde{h}$, and the only $\hat{v}_2 \in \hat{\mathcal{V}}_2$ that supplies a $\tilde{\beta}_{j,\xi}$ factor together with a $\tilde{y}$ factor is $\hat{\delta}_\xi$. Pairing the two yields only terms of the form $\tilde{s}\tilde{y}_{i'} \tilde{y}_k \tilde{\beta}_{k,\xi}$ or $\tilde{r}_{i'} \tilde{y}_k \tilde{\beta}_{k,\xi}$ for $k \in S_\xi$. To match the $\tilde{\beta}_{j,\xi}$ factor we need $k = j$, and then the remaining $\tilde{y}_{i'}$ (resp. $\tilde{r}_{i'}$) factor would have to equal $\tilde{y}_i$ (resp. $\tilde{r}_i$), which forces $i' = i$. In particular, this rules out $i' = j$ as pairing $\hat{c}_{3,j}^*$ with $\hat{\delta}_\xi$ gives $\tilde{s}\tilde{y}_j \tilde{y}_i \tilde{\beta}_{i,\xi}$ (when $i \in S_\xi$), which carries $\tilde{\beta}_{i,\xi}$ rather than $\tilde{\beta}_{j,\xi}$ and is therefore a different monomial. Finally, as argued in Case I-b, $\hat{\beta}_{j,\xi}$ and $\hat{\delta}_\xi$ are the only elements of $\hat{\mathcal{V}}_2$ which induce a $\tilde{\beta}_{j,\xi}$ term, and the former would have to be paired with an element of $\hat{\mathcal{V}}_1$ inducing $\tilde{s}\tilde{y}_i \tilde{y}_j$ or $\tilde{r}_i \tilde{y}_j$, of which there are none, so no other pairing can induce the required $\tilde{y}_j \tilde{\beta}_{j,\xi}$ factor either. We note in particular that $\tilde{\beta}_{j,\xi}$ is unique to the key ξ , so these terms cannot be cancelled by $\hat{c}_{3,i}^* \hat{\delta}_{\xi'}$ terms for any $\xi' \neq \xi$. Note that, like Case III-a, a $\hat{c}_1^*$ term induces neither an $\tilde{s}\tilde{y}_i \tilde{y}_j \tilde{\beta}_{j,\xi}$ nor an $\tilde{r}_i \tilde{y}_j \tilde{\beta}_{j,\xi}$ term because $\langle m_b^* \rangle$ must have been created before the challenge phase and neither $\tilde{s}$ nor any $\tilde{r}_i$ appears in a $\tilde{\mathcal{T}}$ entry before then.

Summarizing the analysis above, we have

$$\begin{aligned}
\text{coeff}_{\tilde{s}\tilde{y}_i}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^*}(\hat{h}) && \text{for } i \in S^* \\
\text{coeff}_{\tilde{r}_i}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^*}(\hat{h}) && \text{for } i \in [L] \setminus S^* \\
\text{coeff}_{\tilde{s}\tilde{y}_i \tilde{u}_j}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{u}_j}(\hat{h}) && \text{for } (i, j) \in S^* \times [K], \\
\text{coeff}_{\tilde{r}_i \tilde{u}_j}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{u}_j}(\hat{h}) && \text{for } (i, j) \in ([L] \setminus S^*) \times [K], \\
\text{coeff}_{\tilde{s}\tilde{y}_i \tilde{y}_j \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\delta}_\xi}(\hat{h}) && \text{for } (i, \xi) \in S^* \times [q_{\text{KG}}], j \in S_\xi, \\
\text{coeff}_{\tilde{r}_i \tilde{y}_j \tilde{\beta}_{j,\xi}}(\tilde{h}) &= \text{coeff}_{\hat{c}_{3,i}^* \hat{\delta}_\xi}(\hat{h}) && \text{for } (i, \xi) \in ([L] \setminus S^*) \times [q_{\text{KG}}], j \in S_\xi.
\end{aligned}$$

It follows that if any of the Case III terms are included in $\hat{h}$, then $\tilde{h}$ has a corresponding term and thus $\tilde{h}$ is never identically zero. Thus,

$$\Pr[\text{bad}_{\text{III}}] = 0.$$

Combining the case analysis above, we overall have that

$$\begin{aligned}
\Pr[\text{bad}] &\leq \Pr[\text{bad}_{\text{I-a}}] + \Pr[\text{bad}_{\text{I-b}}] + \Pr[\text{bad}_{\text{II}}] + \Pr[\text{bad}_{\text{III}}] \\
&= \text{negl}(\lambda).
\end{aligned}$$

□

Lemma A.17. For all $\lambda \in \mathbb{N}$ and all $\chi \in [q_{\text{ZT}}]$, $\Pr[\text{Hyb}_{5,\chi}^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,\chi,\text{int}}^{(b)}(\lambda)]$.

Proof. All that differs between the two hybrids is when the challenger samples the set S^* and performs the associated substitution on the entries of $\tilde{\mathcal{T}}$. Specifically, this step is performed immediately after the $(\chi - 1)^{\text{th}}$ post-challenge zero-test query in hybrid $\text{Hyb}_{5,\chi}^{(b)}$, while it is instead performed immediately after the χ^{th} post-challenge zero-test query in hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$. We will argue that $\mathcal{A}$'s view is identical in both hybrids.

Throughout, we let SubS_{S^*} denote the ring homomorphism as defined in hybrid $\text{Hyb}_{5,0,\text{int}}^{(b)}$, so that the substitution step performed in both hybrids replaces each entry $\tilde{h}$ of $\tilde{\mathcal{T}}$ with $\text{SubS}_{S^*}(\tilde{h})$. We also recall that, since hybrid $\text{Hyb}_4^{(b)}$, the zero-test oracle answers a query on a handle $\langle h \rangle$ by testing whether $\tilde{h} \equiv 0$, and so the set S^* sampled in this step influences $\mathcal{A}$'s view only through the entries of $\tilde{\mathcal{T}}$.

We first observe that the first $\chi - 1$ post-challenge zero-test queries are answered identically in both hybrids, as the substitution step is performed after the $(\chi - 1)^{\text{th}}$ query in either case.

Next, consider the χ^{th} post-challenge zero-test query on a handle $\langle h \rangle$ with associated adversary formal polynomial $\hat{h} = \tilde{\mathcal{T}}(\langle h \rangle)$ and challenger formal polynomial $\tilde{h}' = \tilde{\mathcal{T}}(\langle h \rangle)$ (where the latter is prior to any substitution). As the behavior added in hybrid $\text{Hyb}_{5,\chi}^{(b)}$ applies to this query, if $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$ or $\text{coeff}_{\hat{c}_{3,i}^*}(\hat{h}) \neq 0$ for some $(i, \hat{c}) \in [L] \times (\{1\} \cup \hat{\mathcal{V}}_2)$, then both hybrids return “non-zero” without accessing the table $\tilde{\mathcal{T}}$ at all, and the query is answered identically. Otherwise, $\text{coeff}_{\hat{c}_1^*}(\hat{h}) = \text{coeff}_{\hat{c}_{3,i}^*}(\hat{h}) = 0$ for all $(i, \hat{c}) \in [L] \times (\{1\} \cup \hat{\mathcal{V}}_2)$. By inspection of Eq. (A.3), $\hat{c}_{3,i}^*$ are the only variables in $\hat{\mathcal{V}}$ that can induce a $\tilde{r}_i$ variable in $\tilde{h}'$, and thus $\tilde{h}'$ contains no monomial involving any of the variables $\{\tilde{r}_i\}_{i \in [L]}$. It follows that $\text{SubS}_{S^*}(\tilde{h}') = \tilde{h}'$, and so the χ^{th} query is again answered identically whether or not the substitution has already been applied.

It remains to argue that the entries of $\tilde{\mathcal{T}}$ coincide in both hybrids from the point immediately following the χ^{th} query onwards, from which it follows that the remaining post-challenge zero-test queries are also answered identically. Consider any handle $\langle h \rangle$ present in $\tilde{\mathcal{T}}$ at this point and let $\tilde{h}'$ again denote the entry it would have in the absence of any substitution. In hybrid $\text{Hyb}_{5,\chi,\text{int}}^{(b)}$, the entry is $\text{SubS}_{S^*}(\tilde{h}')$, so it suffices to show that the same holds in hybrid $\text{Hyb}_{5,\chi}^{(b)}$, which we do by induction on the order in which handles are created.

For the base case, a handle created before the substitution step has entry $\text{SubS}_{S^*}(\tilde{h}')$, as the step is applied to every handle in $\tilde{\mathcal{T}}$ at that time. For the inductive step, consider a handle $\langle h \rangle$ created after the substitution point and suppose that every handle $\langle h_0 \rangle$ created before it has entry $\text{SubS}_{S^*}(\tilde{h}_0')$. We go over each way in which $\langle h \rangle$ may have been produced below:

- **Post-challenge key-generation queries:** On the ξ^{th} key-generation query, the challenger adds the entries $\langle \beta_{i,\xi} \rangle \mapsto \tilde{\beta}_{i,\xi}$ for each $i \in S_\xi$ and $\langle \delta_\xi \rangle \mapsto \tilde{\alpha} \sum_{i \in [K]} \text{id}_\xi^{i-1} \tilde{u}_i + \sum_{i \in S_\xi} \tilde{y}_i \tilde{\beta}_{i,\xi}$ to $\tilde{\mathcal{T}}$. Neither entry contains any of the variables $\{\tilde{r}_i\}_{i \in [L]}$ and so both are unaffected by the SubS_{S^*} ring homomorphism.
- **Post-challenge generic group oracle queries:** The analysis of the generic group oracle queries is as follows:
 - **BGGM.Encode(x, v):** The challenger adds the entry $\langle h \rangle \mapsto x$ to $\tilde{\mathcal{T}}$ where x denotes the constant polynomial, which is again unaffected by the SubS_{S^*} ring homomorphism.
 - **BGGM.Add($\langle h_1 \rangle, \langle h_2 \rangle$):** Recall that the challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that SubS_{S^*} is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) + \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{S^*}(\tilde{h}_1') + \text{SubS}_{S^*}(\tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}_1' + \tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}').$$

- **BGGM.Pair($\langle h_1 \rangle, \langle h_2 \rangle$):** The challenger adds the entry $\langle h \rangle \mapsto \tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle)$ to $\tilde{\mathcal{T}}$. By the induction hypothesis and the fact that SubS_{S^*} is a ring homomorphism, we have that

$$\tilde{\mathcal{T}}(\langle h_1 \rangle) \cdot \tilde{\mathcal{T}}(\langle h_2 \rangle) = \text{SubS}_{S^*}(\tilde{h}_1') \cdot \text{SubS}_{S^*}(\tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}_1' \cdot \tilde{h}_2') = \text{SubS}_{S^*}(\tilde{h}').$$

In each case, the entry of $\langle h \rangle$ coincides with $\text{SubS}_{S^*}(\tilde{h}')$, and thus $\mathcal{A}$'s view is identical in the two hybrids. It follows that

$$\Pr[\text{Hyb}_{5,\chi}^{(b)}(\lambda)] = \Pr[\text{Hyb}_{5,\chi,\text{int}}^{(b)}(\lambda)]. \quad \square$$

Lemma A.18. For all $\lambda \in \mathbb{N}$, $\Pr[\text{Hyb}_{5,q_{\text{ZT}},\text{int}}^{(0)}(\lambda)] = \Pr[\text{Hyb}_{5,q_{\text{ZT}},\text{int}}^{(1)}(\lambda)]$.

Proof. The lemma follows from the fact that both hybrids operate independently of the challenge bit b . In particular, any zero-test query dependent on the challenge-message-dependent component (which is the only component dependent on the bit b) must have that $\text{coeff}_{\hat{c}_1^*}(\hat{h}) \neq 0$, and the oracle only ever returns “non-zero” in this case (as the forced-“non-zero” rule applies to every post-challenge zero-test query, and no pre-challenge query can involve $\hat{c}_1^*$). $\square$

The theorem now follows by applying a standard hybrid argument over [Lemmas A.11](#) to [A.18](#). In particular,

$$\begin{aligned} |\Pr[b^* = 1 : b = 0] - \Pr[b^* = 1 : b = 1]| &= |\Pr[\text{Hyb}_0^{(0)}(\lambda)] - \Pr[\text{Hyb}_0^{(1)}(\lambda)]| \\ &\leq 2(3 + q_{\text{ZT}}) \cdot \text{negl}(\lambda) \\ &= \text{negl}(\lambda), \end{aligned}$$

where we have used the fact that $q_{\text{ZT}} = \text{poly}(\lambda)$. $\square$

Parameter instantiation. We now give one example of how to set the parameters of [Construction A.8](#) so that [Theorem A.9](#) yields inverse-polynomial error and the conditions of [Theorem A.10](#) are satisfied. We stress that there is nothing special about our parameter selection; any choice of L , L_{sk} , L_{ct} , and K satisfying the conditions suffices. Take

$$L = K = \lambda \quad , \quad L_{\text{sk}} = \lambda^{\frac{1}{3}} \quad , \quad L_{\text{ct}} = \lambda - \lambda^{\frac{1}{3}}.$$

Per [Theorem A.9](#), we have that [Construction A.8](#) is statistically correct with error

$$\varepsilon(\lambda) = L_{\text{sk}} \left(1 - \frac{L_{\text{ct}}}{L} \right) = \lambda^{\frac{1}{3}} \cdot \frac{\lambda^{\frac{1}{3}}}{\lambda} = \lambda^{-\frac{1}{3}}.$$

Furthermore, L and K are polynomial in λ , $L_{\text{sk}} = \lambda^{\frac{1}{3}} = \omega(\log \lambda)$, and $L - L_{\text{ct}} = \lambda^{\frac{1}{3}} = \omega(\log \lambda)$. Moreover, we have that $K \cdot L_{\text{sk}} = \lambda^{\frac{4}{3}} \geq 2\lambda = \kappa \cdot L$ with $\kappa = 2$ (for all $\lambda \geq 8$). Thus, by [Theorem A.10](#), [Construction A.8](#) is unique-identity semantically secure in the generic bilinear group model.

Lifting to full correctness and security. As discussed in [Remark A.6](#), we can lift unique-identity semantic security to (standard) semantic security by using a PRF on the identity to derive key-generation randomness. Furthermore, [Remark A.4](#) allows us to go from the inverse-polynomial error achieved by [Construction A.8](#) to negligible correctness error. Applying these remarks with the above parameter instantiation, we obtain the following.

Corollary A.19 (IBE from Pairings in the BGM). *Suppose one-way functions exist. Let λ be a security parameter and let GroupGen be a prime-order asymmetric pairing-group generator. There exists an identity-based encryption scheme with identity space $\mathcal{I} = \{\mathcal{I}_\lambda\}_{\lambda \in \mathbb{N}}$ where $\mathcal{I}_\lambda = \{0, 1\}^\lambda$ that is correct with negligible error and semantically secure in the generic bilinear group model.*