

A General Approach to Adaptor Signatures

Nico Döttling¹, Sri AravindaKrishnan Thyagarajan², Pratik Soni³,
Jay Taylor², Hendrik Waldner¹, and Riccardo Zanotto^{1,4}

¹CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
{doettling, hendrik.waldner, riccardo.zanotto}@cispa.de

²The University of Sydney, Sydney, Australia
aravind.thyagarajan@sydney.edu.au, jtay7135@uni.sydney.edu.au

³University of Utah, Salt Lake City, US
psoni@cs.utah.edu

⁴Saarland University, Saarbrücken, Germany

Abstract

Adaptor signatures have emerged as a powerful *contract-minimal* mechanism for fair exchange on blockchains, enabling efficient and privacy-preserving atomic swaps and conditional payments. However, existing adaptor schemes are limited to narrow classes of NP relations (e.g., discrete logarithm secrets) and specific signature schemes, limiting their scope both in terms of constructions and applications.

This work addresses this gap by presenting a new framework that supports arbitrary NP relations and a broad range of signature schemes, significantly extending the reach of adaptor signatures beyond prior works and broadening the design space for adaptor signature constructions. Our framework also yields adaptor signatures that are compatible with today’s blockchain systems. More specifically, we devise two general compilers for adaptor signatures. First, we show how to efficiently lift adaptor signatures from structured languages (such as discrete log relations) to *general NP languages* with the help of degree-2 homomorphic encryption, resulting in adaptors for standard signature schemes and general NP languages. Secondly, we develop a compiler that transforms any signature scheme with a *subliminal channel* into an adaptor signature scheme for general NP languages using *circuit-private fully homomorphic encryption*. Signatures with subliminal channels enable the encoding of witnesses in the signing randomness, and include randomness-recoverable schemes like Schnorr, ECDSA, CL, BBS, as well as salted versions of deterministic signatures like BLS and RSA. Both constructions achieve a novel property called *statement truth privacy*, which guarantees that a buyer learns nothing about the underlying statement, not even its truth, unless the protocol successfully completes.

Contents

1	Introduction	3
1.1	Our Contributions	4
1.2	Related Work	7
2	Technical Overview	9
2.1	Building the baseline construction	9
2.2	Towards the generic construction	11
2.3	Relation to HE-based FAS construction	12
3	Preliminaries	12
3.1	Digital Signatures	13
3.2	NP Languages	15
3.3	Adaptor Signatures	16
3.4	Pseudorandom Encryption	18
3.5	(Public key) Homomorphic encryption	19
3.6	Sigma protocols	21
3.7	Non-interactive Zero-Knowledge Proofs	21
4	Defining Adaptor Signatures with Advertisement	22
5	From Discrete-logarithm to NP Adaptors for Schnorr (and More)	26
5.1	Ciphertext Equality Proofs	27
5.2	Proof-of-Concept Implementation	30
5.3	Formal Analysis	30
5.3.1	Correctness	30
5.3.2	Extractability	31
5.3.3	Seller Fairness	36
5.3.4	Strong Canonical Signing	41
5.3.5	Statement Truth Privacy	42
6	Adaptor Signatures from Subliminal Channels using Circuit-Private FHE	43
6.1	Correctness	46
6.2	Pre-Signature Adaptability	46
6.3	Extractability	48
6.3.1	Proofs of the Hybrid Transitions and Final Bounds.	51
6.3.2	Bounding the relation-witness event E_{rel}	52
6.3.3	Bounding the signature-forgery event E_{sig}	53
6.4	Strong Canonical Signing	56
6.5	Statement truth privacy	59
6.5.1	Proofs of the Statement-Truth-Privacy Hybrid Transitions.	61
A	A Sigma Protocol for $\mathcal{L}_{\text{CE}}$	67

1 Introduction

Fair exchange demands that either both parties obtain their agreed outputs or neither does, which is impossible in the classical two-party setting without a trusted enforcer [15]. Public blockchains partially substitute for such an enforcer: on-chain conditionals (smart contracts) make delivery contingent on evidence of a counterparty’s action, enabling atomic swaps and contingent payments [18, 41] as well as fair exchange of digital goods and computation outputs [22, 48]. However, smart contracts are costly, expose logic and state publicly, and are not uniformly expressive across chains (e.g., Bitcoin [42]), which motivates contract-minimal approaches to fair exchange.

Adaptor signatures in a nutshell. Adaptor signatures or *scriptless scripts* were introduced by Poelstra [44] to realize contract-free atomic exchanges on a blockchain. Let Σ be the signature scheme used for payments and let R be an NP relation. A seller S holds a witness wit for a statement stmt (i.e., $R(\text{stmt}, \text{wit}) = 1$) and wishes to sell it to a buyer B for v coins. The buyer first produces a pre-signature $\tilde{\sigma}$ for a payment transaction that is bound to stmt . Using wit , the seller adapts $\tilde{\sigma}$ into a valid signature σ that authorizes the payment, and from the pair $(\tilde{\sigma}, \sigma)$ the buyer extracts wit . Security requires that (i) without a valid witness no adversary can turn $\tilde{\sigma}$ into a valid signature (unforgeability), and (ii) anyone who knows both $\tilde{\sigma}$ and σ can recover wit (extractability). Together these guarantee fairness: either both parties get their outputs, or neither does. We illustrate this workflow in Figure 1.

Since their inception, adaptor signatures have enabled applications such as payment channels [41], private coin mixing [31, 36], atomic swaps [49], oracle-based payments [39], and functional payments [51], among others. Their appeal lies in efficiency and compatibility: they avoid heavy on-chain logic while retaining atomicity.

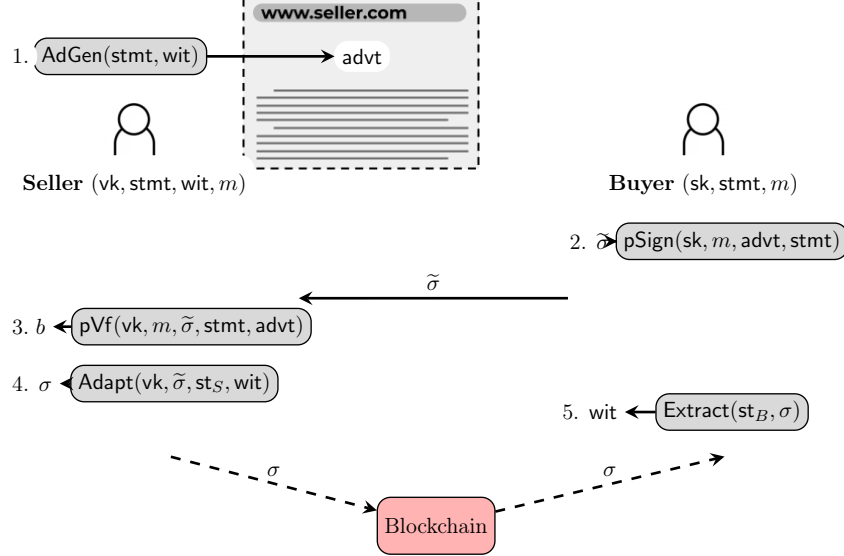


Figure 1: Adaptor signatures for blockchain fair exchange: swapping witness wit and signature σ . The two AdGen and pSign algorithms also output internal states st_S and st_B .

Drop-In Compatibility. A crucial design constraint for adaptor signatures is that the adapted signature must look exactly like a normal signature under the original scheme Σ . In practice this

means we cannot change the signature verification procedure on the blockchain. This requirement makes adaptor design challenging: the hidden witness must be embedded without altering the signature’s format or probability distribution. On the upside, drop-in compatibility means adaptor signatures can deploy immediately on existing blockchains (since consensus rules see no difference). But it also forces designers to find clever algebraic tricks within the existing signature scheme itself. **Existing Constructions.** Despite these constraints, adaptor signatures have been realized for several signature families, including Schnorr signatures, ECDSA, and even post-quantum candidates such as lattice-based Dilithium and isogeny-based CSI-FiSh [25, 47]. Recent work also shows feasibility for certain identification-based signatures that admit homomorphic structure [24]. Gerhart et al. [30] proposed the *dichotomic signatures* framework which broadens coverage to standard-model schemes including CL, BBS+, and Waters+.

Yet progress remains uneven along two axes.

- *Scheme coverage:* many widely deployed or emerging signatures still fall outside known templates, and existing results often hinge on a restricted class of scheme-specific algebraic tricks (e.g., how the nonce or verification equation behaves), offering no reusable recipe.
- *Relation coverage:* most constructions natively support only certain algebraic or homomorphic relations, leaving broader classes of NP relations without a clear path. A few recent works pursue full NP coverage along this axis [16, 38, 14], but do so through dedicated signature mechanisms rather than by preserving the verification algorithm of established or standard schemes. We compare our approach to these constructions in Section 1.2.

Consequently, each new signature or relation requires bespoke engineering and proofs. This motivates our main question:

Can we build adaptor signatures for any NP relation and known signature schemes of interest?

1.1 Our Contributions

Prior work on regular adaptor signatures implicitly assumes that sellers can announce their intent to sell a witness (by showing their public-key and a statement), but this step was never formalized.¹ We make this advertisement phase explicit (Definition 4.1): a seller first publishes a reusable advertisement announcing availability of a witness, before any buyer interaction. This mirrors real deployments (e.g., posting on a bulletin board or website, see Figure 1) and, crucially, is buyer-agnostic and off-chain. Formalizing this phase lets us cleanly separate what is public from what is interactive, and becomes the basis for our later results.

In this model we give two constructions, expanding the class of adaptor constructions along the signature axis and the language axis respectively.

From Dlog- to NP-adaptors. Most existing constructions are restricted to structured languages. For example, Schnorr adaptors are w.r.t. the discrete-logarithm language, which is useful in applications since one can sell secret keys of an ElGamal public key [23]. However, lack of support for general NP relations limits applicability. Thus, given such an adaptor, our first result lifts it to enable sales of general NP witnesses.

¹*Functional adaptor signatures (FAS)* [51], an extension of standard adaptor signatures where the buyer learns a selected function of the witness rather than the witness itself, already incorporate an advertisement-style setup, but this was never formalized for standard adaptor signatures.

In fact, as we show in Section 5, the above idea is not restricted to Schnorr adaptor, but rather extends to any adaptor signature scheme w.r.t. relation R which contains key-pairs of a linearly-homomorphic encryption scheme.²

Theorem (Informal, Theorem 5.1). *Let LE be a linearly homomorphic encryption scheme. Suppose there exists an adaptor signature scheme for signature scheme Σ w.r.t. the LE scheme’s key relation $\{(\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}})\}$. Assume the existence of a circuit-private degree-2 homomorphic encryption scheme.³ Then, for every NP language $\mathcal{L}$, we have an adaptor signature with advertisement scheme AS^{Ad} for $(\mathcal{L}, \Sigma)$.*

Our first compiler is motivated by two new security notions specific to selling via advertisements: *seller fairness* and *distributional statement truth privacy*. Seller fairness (see Definition 4.5) removes the need for explicit pre-signature verification. In standard adaptor signatures, the seller checks that the buyer’s pre-signature is well formed before attempting adaptation; otherwise, the seller aborts and no witness is leaked. Our definition provides the same effective protection without this check: either the pre-signature adapts into a valid payment signature, or failed adaptation attempts reveal no information about the seller’s witness. Looking ahead, this is important for our compiler as proving correct pre-signature generation can be expensive.

On the other hand, *distributional statement truth privacy* (Definition 4.6) addresses a leakage channel that public advertisements create and that prior work never had to consider. In the usual setting the advertisement is itself a proof, or a commitment to a valid witness, so the verdict, whether a witness exists at all, is public from the moment it is posted. Fairness then covers only the exchange: the buyer recovers the witness exactly when the seller is paid. Our advertisement instead commits to selling a witness *without disclosing whether one exists*. Formally, an honest advertisement for a true statement is indistinguishable from a simulated advertisement for a false one, so the advertisement reveals nothing about the statement’s truth beyond what the public statement distribution already reveals. The verdict is settled only by a completed exchange. If the statement is true, the buyer obtains the witness and the seller is paid. If not, the exchange fails, the seller earns nothing, and the buyer pays nothing. The truth of the statement is thus something the buyer pays for, not something the advertisement gives away.

In intended use the seller never advertises a statement it believes false. Either it holds a valid witness and wishes to conceal that fact, or the statement depends on an event that has not yet occurred, so it commits a candidate witness before the truth is determined.

What truth privacy conceals is thus the verdict of a *genuine* offer, not a lie. Because advertisement generation never inspects the witness—validity is enforced at adaptation, not at advertisement. An advertisement is produced identically in every case, and the comparison in Definition 4.6 to a simulated advertisement for a false statement is merely the formal way of saying that a true-statement advertisement carries no verdict: it is indistinguishable from one that encodes no witness at all. Posting a false statement or a bogus witness, moreover, gains the seller nothing—such an advertisement cannot be adapted into a payment signature, so the seller collects nothing and an honest buyer never pays for a non-existent witness. Reputational harm from being caught is therefore beside the point: there is no profitable false advertisement to deter.

This is meaningful in precisely those applications where the verdict is itself the valuable good, or where the seller cannot even know the verdict when it posts.

²For example, Dilithium adaptor is w.r.t. the SIS relation where the witness can be viewed as a secret-key for an appropriate LWE-based linearly homomorphic scheme [51].

³In terms of assumptions, degree-2 homomorphic encryption can be based on hard problems over bilinear maps [6].

- *Loan liquidation.* The seller here is a lender that keeps its loan records private. When a loan’s collateral falls below the debt, a third party may close it out for a fee. That party is the buyer. Before any relevant price is known, the lender posts an advertisement binding a signed loan record $\text{wit} := (\text{rec}, \tau)$, where τ is the lender’s signature on rec under its key vk_L . A price feed oracle later publishes a price e . The buyer sets $\text{stmt} := (\text{vk}_L, e)$, and the relation holds when $\Sigma.\text{Vf}(\text{vk}_L, \text{rec}, \tau) = 1$ and the collateral in rec is worth less than the debt at price e . The buyer cannot forge τ , so the relation stays hard even though e is public. On success it decrypts rec and learns which loan to close and on what terms. The lender cannot know at posting time which loans will qualify, since the price e does not yet exist. Once e is published by the oracle, anyone can pair it with the advertisement. Truth privacy is what stops them from learning, without paying, whether the loan can be closed out.
- *Compliance screening.* A provider of address-risk data maintains a private list of flagged addresses and sells signed evidence of membership. For each address A it covers, whether or not it is listed, the provider posts an advertisement that asserts nothing about A ’s status. A buyer sets $\text{stmt} := (\text{vk}_P, A)$, and the relation holds when the advertisement binds a signature $\text{wit} := \sigma$ satisfying $\Sigma.\text{Vf}(\text{vk}_P, A, \sigma) = 1$, that is, when the provider has attested that A is listed. The buyer cannot forge σ , so the relation is hard, and since an address reveals nothing about its status, true and false statements are indistinguishable a priori. On success the buyer decrypts and obtains σ itself. A regulated firm must retain the signed attestation for its auditor and its supervisor, so it holding the evidence that A is flagged is the product. Were the advertisement to reveal membership, the list would be public for free and the service would have nothing left to sell.

To the best of our knowledge, prior adaptor-signature constructions either assume that the advertised statement is true or reveal it through the advertised proof itself. We formalize seller fairness and distributional statement truth privacy, prove them for our construction, and report a proof-of-concept implementation, in Section 5.

We next turn to how broadly adaptor signatures can be realized across signature schemes.

Adaptors for Signatures with Subliminal Channels. Past constructions have largely relied on ad-hoc algebraic tricks, most often, embedding the secret witness into the *randomness* used during signing. While effective in specific cases, this approach ties adaptor design to the specific algebraic structure of individual signature schemes, requiring more concrete properties and not allowing for general extensions. Our work abstracts this idea into a more principled notion. We observe that many signature schemes satisfy *randomness recovery*, where the signer can deterministically reconstruct the signing randomness from a produced signature. Randomness recovery immediately enables a subliminal communication channel⁴: one can encrypt a message, use the ciphertext as signing randomness⁵, and later recover and decrypt it. We generalize this observation into a broader class of schemes, which we call *signatures with subliminal channels*, which captures both randomness-recoverable schemes and those where randomness can only be partially or indirectly extracted.

Theorem (Informal, Theorem 6.1). *Assume the existence of circuit-private fully homomorphic encryption FHE, and a secret key encryption scheme SKE with pseudorandom ciphertexts. Then, for*

⁴Anamorphic signatures [37] similarly involves hiding an “anamorphic message” inside the signature, which an extractor knowing the double-key (akin to a subliminal extractor) can recover.

⁵This, additionally, requires the ciphertext to look random.

any NP language $\mathcal{L}$ and any strongly unforgeable signature scheme Σ with subliminal channels, there exists an adaptor signature scheme with advertisement AS^{Ad} for $(\mathcal{L}, \Sigma)$.

At a high level, the construction uses the signature itself as the delivery channel for the witness. The witness is encrypted under a symmetric key held by the buyer, the resulting ciphertext is encoded as the signing randomness, and a standard signature is generated using it. The seller receives a valid payment signature, while the buyer recovers the ciphertext by subliminal extraction and decrypts it to obtain w .

Given that constructing the final adapted signature in the described manner requires a joint computation, by the buyer and the seller, we use circuit-private fully homomorphic encryption (FHE) [8, 21, 29] to realize this. More precisely, the advertisement will consist of a FHE encryption of a witness. The presignature is then generated by generating a pseudorandom encryption under the FHE scheme and use the obtained ciphertext as the randomness in the signing procedure. This computation only succeeds if the witness encrypted in the advertisement is valid. The presignature can then be adapted by the seller by simply decrypting the received presignature. In the advertisement generation as well as in the presignature generation NIZK proofs are used to assert the correctness of the computation.

This construction satisfies distributional statement truth privacy and the standard pre-signature adaptability property. Unlike our first compiler, it therefore retains an explicit pre-verification step; the gain is that it applies to a much *broader class* of signature schemes. This expands the supported language to all of NP. Even more importantly, our result significantly broadens the set of signature schemes for which adaptors can be built (with drop-in compatibility).

- Our framework of randomness recoverable signatures already includes schemes like Schnorr, ECDSA, CL, and BBS (Definition 3.5) where we can extract the full randomness used during signature generation. In fact, our compiler also works for a relaxation of randomness recoverable signatures - *subliminal signatures* (see Definition 3.6). This relaxes the randomness recovery requirements to only be able to extract the encoded witness. For example, the strongly unforgeable variant of Waters signatures scheme [7] falls into this category.
- Interestingly, constructing adaptor signatures for deterministic schemes has remained elusive. This is not accidental but inherent: Erwig et al. [24] showed that unique signatures cannot admit adaptors for hard NP languages. Hence, randomness is essential.

A natural and widely used way to introduce randomness is *salting* [35, 4]. Salting augments the signature with random coins while leaving the verification algorithm unchanged—and crucially, it yields a randomness-recoverable scheme. This immediately gives adaptor constructions for salted versions of all deterministic and unique signature schemes. For example, our result enables fair exchange even for a unique signature scheme like BLS, which can be used in Ethereum for verification of transactions (with salts).

1.2 Related Work

General NP relations for adaptor signatures. Dai, Okamoto, and Yamamoto [16] construct adaptor signatures from any signature scheme and any hard relation. To do so they explicitly revisit the scriptless-scripts constraint, replacing Σ with an *augmented* scheme whose verification additionally checks relation membership, so the adapted signature carries the witness in the clear. They note that implementing this on existing blockchains amounts to using scripts. Liu, Tzannetos, and Zikas [38] observe that exposing the witness undermines the target applications, introduce *witness*

hiding, and repair the construction with a weak variant of trapdoor commitments instantiated from the Hamiltonian cycle problem through a Karp reduction. Ciampi, Liu, Tzannetos, and Zikas [14] replace the Karp reduction with MPC in the head (MPCitH) and note in turn that validating their signatures requires a Turing-complete scripting language and that Bitcoin cannot verify their scheme.

All three follow the same template: the posted object is a tuple carrying the instance alongside auxiliary material, and verification invokes Σ on a message encoding that instance, rather than on the transaction itself, before applying a further check, respectively of relation membership, of a commitment opening, and of an MPCitH proof. They are unfortunately not signatures that current transaction verifiers for Σ would accept. This matters because adaptor signatures owe their appeal to drop-in compatibility with consensus rules that are already fixed. Gerhart et al. [30] argued this point for constructions that target new *signature schemes*, observing that prior work either restricts to identification schemes converted through Fiat-Shamir or else modifies verification. The constructions above target new *relations* and give up the same guarantee. In contrast, our constructions extend coverage in both directions while keeping it, since the adapted signature verifies under the unmodified algorithm of Σ .

The Advertisement Model. Standard adaptor definitions [25, 2, 24, 38, 16] formalize pre-signing relative to a public statement X (and some auxiliary information in the case of [30]), while leaving its generation and publication outside the primitive’s syntax. Vanjani, Soni, and Thyagarajan [51] make this seller-initiated phase explicit for functional adaptor signatures through advertisement generation and verification algorithms and by requiring the new security property of advertisement soundness. We develop the analogous model for standard adaptor signatures and introduce two security notions tailored to public, buyer-agnostic advertisements: seller fairness and distributional statement truth privacy. Relatedly, Della Monica et al. [19] study advertisements at the level of complete fair-exchange protocols, where sellers publish verifiable previews before exchange. Our focus is instead primitive-level, isolating the advertisement interface and its security requirements within adaptor signatures for modular use in higher-level applications.

Zero-knowledge contingent payments and witness encryption. Our advertisement model is related to zero-knowledge contingent payments (ZKCP) [40], where a seller commits to a digital good off-chain and a buyer releases payment once convinced, through a proof, that the commitment matches an agreed statement.⁶ Payment settles through an on-chain hash-lock rather than a plain signature, the gap scriptless scripts (or adaptor signatures) was introduced to close and which we retain by not requiring a new signature verification algorithm. The resemblance to our work is closest in the off-chain phase, and there the ZKCP message is generated per buyer, whereas our advertisement is published once and is buyer-agnostic. Witness encryption [28] offers an alternative route to a similar goal, encrypting a message under a statement so that only a satisfying witness decrypts it, but it is not known for general NP from standard assumptions and remains a feasibility result. Our first compiler instead uses only black-box group-based tools and is backed by a proof-of-concept implementation (Section 5).

Statement truth privacy and InfoCP. Nguyen, Ambrona, and Abe [43] formalize what a ZKCP message may leak as *InfoCP*: with the common reference string chosen adversarially by the buyer, a real message encrypting the good is indistinguishable from a simulated one encrypting zeros. In both

⁶The security of ZKCP has been refined repeatedly: Campanelli et al. [10] showed the original proposal can leak through a maliciously chosen common reference string, Fuchsbaauer [27] showed that witness indistinguishability alone is insufficient, and Nguyen, Ambrona, and Abe [43] revisited the question, giving a witness indistinguishable instantiation under a strengthened notion.

cases the proved statement lies in the language, and the predicate the good must satisfy is public and fixed, so what InfoCP hides is the *value* the seller holds. Our distributional statement truth privacy (Definition 4.6) compares an honest advertisement for a statement drawn from a distribution of true statements against a simulated advertisement for one drawn from a distribution of false statements, and so hides whether the statement lies in the language at all. This is meaningful because our statement is externally determined rather than manufactured by the seller, for instance whether an event occurred or whether a record exists. Both notions are simulation-based and we see ours as continuing this line, but neither implies the other. Accordingly our buyer is protected not by an up-front certificate of validity, which truth privacy forbids, but by the guarantee that a false advertisement cannot be adapted into a valid payment signature.

Relation to functional adaptor signatures. Functional adaptor signatures (FAS), introduced by Vanjani, Soni, and Thyagarajan [51] and recently extended to non-linear functions via homomorphic encryption [50], let a buyer learn a function $f(x)$ of the seller’s secret rather than the secret itself. The sharpest separation concerns what the advertisement may reveal. FAS’s advertisement soundness requires that no advertisement for a false statement pass verification, so a passing advertisement is by definition a public attestation that the statement is true. Distributional statement truth privacy demands the opposite, and an advertisement that hides whether a witness exists cannot also certify that one does. Truth privacy is therefore not merely absent from existing FAS constructions but incompatible with FAS’s advertisement soundness, placing the lottery and oracle attestation sales of Section 1.1 beyond its reach by construction.

Three further separations persist even if one specializes FAS to the identity function, where the buyer learns the whole witness. First, in FAS the buyer chooses f after seeing the advertisement, so the seller must send a fresh message for every sale, whereas our advertisement is published once and any buyer can pre-sign directly from it in a non-interactive manner (as required by standard adaptor signatures). Second, FAS attaches a general-purpose NIZK to its advertisement certifying that the encrypted value is a valid witness, whereas we keep only a plaintext-equality proof and a proof that the public key is well formed. Dropping the certificate is what makes truth privacy possible, and it changes how each party is protected: the buyer now homomorphically evaluates the NP verifier itself, and the seller, receiving a masked pre-signature rather than a plain one, is protected by *seller fairness*, a real-versus-ideal relaxation of adaptability under which failed adaptations leak nothing. Third, our second compiler covers any signature admitting a subliminal channel, whereas the FAS constructions are specific to Schnorr. Additionally, the two lines also differ in how they use homomorphic encryption, despite a superficially similar template. We defer that comparison to Section 2.3, once the overview of our mechanism is in place.

2 Technical Overview

We will briefly give a high-level overview of the main ideas behind our adaptor signature constructions. The relevant security notions were discussed informally in the introduction and are formalized in Section 4; here we focus on the technical mechanisms behind our constructions.

2.1 Building the baseline construction

We begin with the simplest setting: an adaptor signature scheme for the discrete-logarithm relation. Suppose a seller wants to sell a witness `wit` for an NP statement `stmt`. A natural first idea is to reduce

this to the sale of a decryption key. The seller samples a fresh secret z , publishes the corresponding ElGamal public key $\mathbf{pk} = g^z$, and encrypts $\mathbf{wit}$ under $\mathbf{pk}$, obtaining a ciphertext c . The advertisement is then $(\mathbf{stmt}, \mathbf{pk}, c)$.

A buyer who wants to obtain $\mathbf{wit}$ produces a pre-signature $\tilde{\sigma}$ with respect to the public key $\mathbf{pk}$. Using the discrete logarithm z , the seller adapts $\tilde{\sigma}$ into a valid payment signature σ . Once σ is released, the buyer extracts z from the pair $(\tilde{\sigma}, \sigma)$ and uses it to decrypt c , thereby recovering $\mathbf{wit}$.

This already captures the basic compiler idea: an adaptor signature for selling a discrete logarithm can be used to sell any witness encrypted under the corresponding public key. However, the naïve approach gives the buyer no assurance that the encrypted value is actually a valid witness for $\mathbf{stmt}$. A malicious seller could encrypt an arbitrary string and still obtain a payment signature. One generic fix would be to attach a NIZK proof that c encrypts a valid witness for $\mathbf{stmt}$. But this has two drawbacks: it relies on heavy non-black box techniques, and, more importantly for us, it reveals that the advertised statement is true. This is exactly the leakage ruled out by distributional statement truth privacy.

Our construction avoids proving validity in the advertisement. Instead, the advertisement contains two encryptions of the same encoded witness: an ElGamal ciphertext c under $\mathbf{pk}$, and a ciphertext c' under a degree-2 homomorphic encryption scheme, concretely instantiated with BGN [6]. The seller also provides an efficient black-box NIZK proof π showing plaintext equivalence between c and c' . Crucially, π proves only that the two ciphertexts encrypt the same value; it does not prove that this value is a valid witness. Thus, the advertisement remains compatible with statement truth privacy. We construct such plaintext-equivalence proofs, which may be of independent interest, in Appendix A.

It remains to explain how the buyer can protect itself from the advertisement, without learning whether $\mathbf{stmt}$ is true. To achieve this, we rely on the homomorphic ciphertext c' . By augmenting the witness with the internal wire values of the NP verification circuit, we can express verification as a quadratic relation [34]. This allows the buyer to homomorphically evaluate the verifier using a degree-2 scheme such as BGN. For notational convenience, we use the convention that the verifier outputs 0 for accept and 1 for reject.

After checking the plaintext-equivalence proof π , the buyer homomorphically evaluates the verifier on c' . This yields a ciphertext c'' encrypting a bit μ , where $\mu = 0$ if the encrypted witness is valid and $\mu = 1$ otherwise. The buyer then samples a symmetric key k_L and a random mask ρ , and computes a ciphertext

$$c^* = \rho \cdot c'' + \text{Enc}(k_L),$$

where the operations are homomorphic. Thus, if $\mu = 0$, then c^* encrypts k_L ; but if $\mu = 1$, then c^* encrypts $k_L + \rho$, which is statistically independent of k_L . The buyer also encrypts its pre-signature under k_L , obtaining $c_{\tilde{\sigma}}$, and sends $(c^*, c_{\tilde{\sigma}})$ to the seller.

If the encrypted witness is valid, the seller decrypts c^* to recover k_L , decrypts $c_{\tilde{\sigma}}$ to obtain $\tilde{\sigma}$, and adapts it into a valid signature σ . The buyer then extracts the ElGamal secret key z from $(\tilde{\sigma}, \sigma)$ and finally decrypts c to recover $\mathbf{wit}$. If the encrypted witness is invalid, then the seller decrypts c^* to a random value independent of k_L . Consequently, the symmetric encryption hides $\tilde{\sigma}$, and the seller cannot obtain a valid payment signature. This gives the buyer protection against malicious advertisements without requiring the advertisement itself to prove that the statement is true.

The same mechanism also preserves statement truth privacy. The advertisement contains only semantically secure encryptions and a plaintext-equivalence proof between them. It does not contain a proof that the encrypted witness is valid for the statement. Therefore, for indistinguishable

distributions of true and false statements, the advertisement does not reveal which case holds; the truth value is only resolved if the protocol completes.

In order to protect an honest seller, who is trying to get paid in exchange of the knowledge of the witness, from a malicious buyer, we introduce the notion of seller fairness. This is a relaxation of the very strong adaptability notion from standard, non-interactive adaptor signatures. Since our adaptor signature with advertisement is inherently an interactive protocol, we borrow ideas from the MPC setting and define our security notion as a real-vs-ideal property. In particular, we model adversaries that send pre-signatures $\tilde{\sigma}$ to the seller, and observe the result: if there is no final signature posted on the blockchain, it means that the **Adapt** algorithm failed. Seller fairness prevents this side leakage from becoming an actual source of information about the hidden witness.

We realize our notion again by exploiting specific features of the homomorphic encryption we use (in particular, BGN). The main problem would be that we need to decrypt ciphertexts coming from the adversary: recall that the first step of the **Adapt** procedure is to decrypt c^* to (hopefully) get k_L , if the witness was valid. However, given the homomorphic nature of our scheme, this could lead into CCA-style problems. To overcome this issue, in a first pass we require our homomorphic evaluation to be deterministic, and then make the randomness ρ used in masking depend on the secret k_L we’re hiding itself. Finally, the most important ingredient is the possibility of *re-randomizing* a ciphertext, meaning that we can “add encryptions of 0” in order to make a ciphertext look completely indistinguishable from a fresh one.

Concretely, the masked ciphertext actually becomes

$$c^* = H_1(k_L) \cdot c'' + \text{Enc}(k_L; 0) + \text{Enc}(0; H_2(k_L)),$$

where everything is deterministic, except the mask $\rho = H_1(k_L)$ and the re-randomization randomness $H_2(k_L)$. Now the seller can do a “CCA-prevention check” by first decrypting c^* , and then re-encrypting it to check that they get the same exact c^* : given the large amount of possible ciphertexts, the only way for an adversary to make this procedure go through is to know the randomness $H_2(k_L)$, i.e. knowing in advance to which value the ciphertext would decrypt.

This means that we can simulate a seller, since from having access to the H_2 random oracle we have a way to decrypt ciphertexts, and we can then simply reduce to the IND-CPA security of the encryption schemes in the advertisement.

A critical feature of our construction is that it avoids inefficient non-black-box techniques and relies only on efficient, black-box primitives in cryptographic groups. We note that this template works generically for any type of homomorphic encryption schemes that fulfill certain properties. Therefore, we prove it more generically, abstracting BGN as HE and ElGamal as LE in the main body of this work.

2.2 Towards the generic construction

We construct adaptor signatures for general NP relations from a digital signature scheme with a subliminal channel and circuit-private FHE, relying on NIZKs and a secret-key encryption scheme with pseudorandom ciphertexts. The main idea is to rely on the FHE scheme to execute an oblivious signature generation that encodes the witness inside a pseudorandom encryption scheme whose randomness is used to generate the mentioned (encrypted) signature. The resulting ciphertext can then be decrypted to obtain such that seller can obtain the final signature.

As already mentioned in the introduction, we also present a generic realization of adaptor signatures for all NP relations relying on circuit-private FHE, signatures with subliminal channels,

pseudorandom encryption and non-interactive zero-knowledge proofs. In this scheme, the advertisement consists of an FHE ciphertext, encrypting the witness, as well as a NIZK proof asserting its correct generation. If a buyer wants to obtain the knowledge of the witness, it can generate the pre-signature by sampling a key for the pseudorandom encryption scheme and use it to encrypt the witness under the FHE scheme. It then uses the obtained ciphertext, again under the FHE encryption, to use it as the randomness for the signature generation using the desired message m . All of this happens under the condition that the encrypted witness actually satisfies the relation. In the last step, another NIZK proof is attached that certifies the correct generation of the signature. This proof also directly achieves pre-signature adaptability; the pre-signature adaptability procedure simply consists of verifying this NIZK proof. The adapt procedure of the seller then corresponds to simply decrypting the FHE ciphertext to obtain the desired signature. When the buyer obtains the signature, it can extract its randomness, which corresponds to the ciphertext encrypting the witness, and use the key of the pseudorandom encryption scheme to decrypt the mentioned ciphertext and obtain the witness. We present this scheme together with its proof formally in Section 6.

2.3 Relation to HE-based FAS construction

Our first compiler and the HE-based FAS constructions [50] combine homomorphic encryption, an adaptor-compatible encryption scheme, and a cross-scheme proof of plaintext equality. We do not regard this shared pattern as a point of separation; what differs is how it is used. FAS evaluates the buyer’s function f and the seller opens the result, so its equality proof is produced once per sale and links a BGN commitment to $f(x)$ to a fresh ElGamal encryption of the same value. We evaluate the *NP verifier* down to a single accept/reject bit that masks the base pre-signature, and our equality proof is produced once inside the reusable advertisement, linking two encryptions of the witness so that the ciphertext the buyer checks is the one it later decrypts. That is precisely what protects the buyer without revealing statement truth.

The two uses of BGN accordingly impose opposite requirements. Their BGN instantiation never decrypts, treats BGN as a commitment, and removes its blinding terms so that evaluation is deterministic and randomness cheaply recoverable. We decrypt during adaptation, so we need BGN as an encryption scheme whose secret key the seller holds, and beyond deterministic evaluation we require the ability to re-randomize both ciphertexts and messages, which is what underlies our masking and seller-fairness arguments. In fact, in their case a seller holding the BGN secret key would break the soundness of their equality proof, so their key must come from a trusted setup, whereas ours is generated by the seller itself with a lightweight well-formedness proof and needs no such trusted key generation. And our plaintext-equality argument is generic across schemes bi-homomorphic in message and randomness and is proved knowledge-sound over bit plaintexts (so that NP verification is expressible as a quadratic), rather than the bounded-integer, special-sound Σ -protocol with abort that FAS tailors to a single ElGamal–BGN pair.

3 Preliminaries

We first introduce some notations used in the paper. By $x \leftarrow \$ X$, we denote the uniform sampling of x from the set X , and λ represents the security parameter. By $y \leftarrow A(x; r)$ we denote a probabilistic polynomial time (PPT) algorithm A that on input x and randomness r , outputs y . When A is a deterministic polynomial time (DPT) algorithm, we use the notation $x := A(y)$. We call a function

$\nu : \mathbb{N} \rightarrow \mathbb{R}$ negligible in n if for every $k \in \mathbb{N}$, there exists $n_0 \in \mathbb{N}$ s.t. for every $n \geq n_0$ it holds that $|\nu(n)| \leq n^{-k}$. We use the function `assert` to check if a condition holds. The function call `assert condition` aborts the game in which it is called if `condition` evaluates to `false`. We briefly recall the main tools need for Dlog to general NP adaptors transformation. A detailed formal introduction of those primitives, as well as the primitives needed for the construction of adaptor signatures from subliminal channels can be found in Section 3.

3.1 Digital Signatures

We recall the syntax of digital signatures.

Definition 3.1 (Digital signature). *Let $\mathcal{D}_{\text{SK}}$ be the secret key space, $\mathcal{D}_{\text{PK}}$ be the public key space, $\mathcal{D}_{\text{R}}$ be the randomness space, $\mathcal{D}_{\text{S}}$ be the signature space and $\mathcal{D}_{\text{M}}$ be the message space. A digital signature scheme $\Sigma = (\text{KGen}, \text{Sign}, \text{Verify})$ for messages of length ℓ_m is a triple of efficient algorithms defined as:*

- $(\text{sk}, \text{vk}) \leftarrow \text{KGen}(\lambda)$. *The key generation algorithm $\text{KGen}(1^\lambda)$ is a PPT algorithm that on input a security parameter 1^λ , outputs a key pair $(\text{sk}, \text{vk}) \in \mathcal{D}_{\text{SK}} \times \mathcal{D}_{\text{PK}}$.*
- $\sigma \leftarrow \text{Sign}(\text{sk}, m; r)$. *The input of the PPT signing algorithm $\text{Sign}(\text{sk}, m)$ is a private key $\text{sk} \in \mathcal{D}_{\text{SK}}$, a message $m \in \{0, 1\}^{\ell_m} = \mathcal{D}_{\text{M}}$, and randomness $r \in \{0, 1\}^{\ell_r} = \mathcal{D}_{\text{R}}$, it outputs a signature $\sigma \in \mathcal{D}_{\text{S}}$.*
- $b \leftarrow \text{Verify}(\text{vk}, m, \sigma)$. *The verification algorithm $\text{Verify}(\text{vk}, m, \sigma)$ is a DPT algorithm that takes as input a public key $\text{vk} \in \mathcal{D}_{\text{PK}}$, a message $m \in \{0, 1\}^{\ell_m} = \mathcal{D}_{\text{M}}$ and a signature $\sigma \in \mathcal{D}_{\text{S}}$ and outputs a bit b .*

We define the correctness of signatures in the usual way:

Definition 3.2 (Correctness). *A digital signature scheme Σ is correct if for any $m \in \{0, 1\}^{\ell_m}$ it holds, that for all $\lambda \in \mathbb{N}$ and all $(\text{sk}, \text{vk}) \leftarrow \text{KGen}(1^\lambda)$:*

$$\Pr[\text{Verify}(\text{vk}, m, \text{Sign}(\text{sk}, m)) = 1] = 1,$$

where the randomness is taken over the random choice of KGen and Sign .

Before defining security, we define the notion of *verification key derivation*. This notion allows to reconstruct the verification key given just the signing key.

Definition 3.3 (Verification-key derivation). *A signature scheme $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Verify})$ supports verification-key derivation if there is a deterministic polynomial-time algorithm*

$$\text{vk}/\perp \leftarrow \Sigma.\text{DerVerKey}(\text{sk})$$

such that the following properties hold:

- (i) For every key pair (sk, vk) in the support of $\Sigma.\text{KeyGen}(1^\lambda)$,

$$\Sigma.\text{DerVerKey}(\text{sk}) = \text{vk}.$$

- (ii) For every signing key sk for which $\Sigma.\text{DerVerKey}(\text{sk}) = \text{vk} \neq \perp$, every message m , and every supported choice of signing coins r ,

$$\Sigma.\text{Verify}(\text{vk}, m, \Sigma.\text{Sign}(\text{sk}, m; r)) = 1.$$

To realize this property, we can generically store the key-generation randomness in the signing key. More precisely, if $(\text{sk}_0, \text{vk}) \leftarrow \Sigma.\text{KeyGen}(1^\lambda; \rho)$, output $\text{sk} = (\text{sk}_0, \rho)$. On input (sk_0, ρ) , the derivation algorithm reruns key generation with randomness ρ and outputs the resulting verification key if the recomputed signing key equals sk_0 ; otherwise, it outputs $\perp$. Signing with (sk_0, ρ) simply runs the original signing algorithm using sk_0 .

Condition (i) follows then because rerunning key generation with the honestly stored randomness reproduces vk . Moreover, whenever **DerVerKey** returns a verification key, its check certifies that (sk_0, vk) is an honestly generated key pair. Therefore, condition (ii), directly follows from the correctness of Σ .

Security of Signatures. In the following, we recall the notions of unforgeability and strong unforgeability. In both games, the adversary $\mathcal{A}$ is given as input the public vk and has access to a signing oracle $\mathcal{O}_{\text{Sign}}(\cdot)$ that $\mathcal{A}$ may query adaptively on messages of his choice. The difference between unforgeability and strong unforgeability lies in the success determination of $\mathcal{A}$. In case of unforgeability, the adversary wins the game if it computes a valid signature σ^* on a new message m^* , i.e., one that was not submitted to the signing oracle. The notion of strong unforgeability demands that $\mathcal{A}$ must compute a fresh message-signature pair (m^*, σ^*) to win the game. The difference is that $\mathcal{A}$ can win the game even if it computes a new signature for a message it queried to the signing oracle. Both security notions are defined in the following. We mark the slight differences between both security definitions with a gray background.

Definition 3.4 (EUF-CMA (SUF-CMA) security). *A signature scheme Σ is (strongly) existential unforgeable under a chosen message attack or EUF-CMA (resp. SUF-CMA) secure, if for every PPT adversary $\mathcal{A}$ there exists a negligible function ν such that for all $\lambda \in \mathbb{N}$, $\Pr[\text{SigForge}_{\mathcal{A}, \Sigma}(\lambda) = 1] \leq \nu(\lambda)$, where the experiment $\text{SigForge}_{\mathcal{A}, \Sigma}$ is defined in Figure 2 and the probability is taken over all random choices of all randomized algorithms.*

$\text{SigForge}_{\mathcal{A}, \Sigma}(\lambda)$	$\mathcal{O}_{\text{Sign}}(\text{sk}, m)$
1: $\mathcal{Q} := \emptyset$	1: $\sigma \leftarrow \text{Sign}(\text{sk}, m)$
2: $(\text{sk}, \text{vk}) \leftarrow \text{KGen}(1^\lambda)$	2: $\mathcal{Q} = \mathcal{Q} \cup \{m\}$
3: $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Sign}}}(\text{vk})$	3: $\mathcal{Q} = \mathcal{Q} \cup \{(m, \sigma)\}$
4: return $(m^* \notin \mathcal{Q} \wedge \text{Verify}(\text{vk}, m^*, \sigma^*))$	4: return σ
5: return $((m^*, \sigma^*) \notin \mathcal{Q} \wedge \text{Verify}(\text{vk}, m^*, \sigma^*))$	

Figure 2: Security notions of digital signatures, where the gray part refers to strong unforgeability and the regular part to unforgeability. The gray winning condition replaces the line before.

Randomness Recoverability. We additionally will need the signature scheme to be able to hide information inside the signature itself. Towards this, we define two notions of randomness recovery: Definition 3.5 is a stricter notion that demands full randomness recovery, whereas as Definition 3.6 introduces a relaxation of “partial” recovery that is sufficient for our results [45, 46].

Definition 3.5 (Randomness recovery). *A signature scheme Σ satisfies randomness recovery if there exists a PPT algorithm RandRec such that for any $m \in \{0, 1\}^{l_m}$, it holds that for all $\lambda \in \mathbb{N}$, all $(\text{sk}, \text{vk}) \leftarrow \text{KGen}(1^\lambda)$, and all $\sigma \leftarrow \text{Sign}(\text{sk}, m)$:*

$$\Pr[\sigma = \text{Sign}(\text{sk}, m; \text{RandRec}(\text{sk}, m, \sigma))] = 1,$$

where the probability is taken over the random choices of all algorithms.

Definition 3.6 (Subliminal channel). *A signature scheme Σ has subliminal channels w.r.t. an encryption scheme SKE if there exists a decoding algorithm $\text{SubExt} : \Sigma.\mathcal{D}_{\text{SK}} \times \Sigma.\mathcal{D}_{\text{M}} \times \Sigma.\mathcal{D}_{\text{S}} \times \text{SKE}.\mathcal{D}_{\text{K}} \rightarrow \text{SKE}.\mathcal{D}_{\text{M}}$ such that for any $w \in \text{SKE}.\mathcal{D}_{\text{M}}$ and $m \in \Sigma.\mathcal{D}_{\text{M}}$ it holds that*

$$\Pr \left[\text{SubExt}(\text{sk}, m, \sigma, k) = w \left| \begin{array}{l} (\text{sk}, \text{vk}) \leftarrow \Sigma.\text{KGen}(1^\lambda), \\ k \leftarrow \text{SKE}.\text{KGen}(1^\lambda), \\ r \leftarrow \text{SKE}.\text{Enc}(k, w), \\ \sigma = \Sigma.\text{Sign}(\text{sk}, m; r) \end{array} \right. \right] = 1.$$

Notice that a randomness-recoverable signature easily gives us a subliminal channel via

$$\text{SubExt}(\text{sk}, m, \sigma, k) := \text{SKE}.\text{Dec}(k, \text{RandRec}(\text{sk}, m, \sigma))$$

An example of a signature scheme where the complete randomness is not recoverable but still support subliminal channels is the strongly unforgeable extension of the Waters signature scheme [7].

3.2 NP Languages

We recall basic notation for relations and NP languages that we use throughout.

Relations and NP languages. A (binary) relation Rel is a subset of $\{0, 1\}^* \times \{0, 1\}^*$. We write $(\text{stmt}, \text{wit}) \in \text{Rel}$ to denote that wit is a *witness* for the *statement* stmt with respect to Rel .

Definition 3.7 (NP relation and language). *A relation $\text{Rel} \subseteq \{0, 1\}^* \times \{0, 1\}^*$ is an NP relation if there exists a deterministic polynomial-time algorithm V_{Rel} (the verifier) and a polynomial $p(\cdot)$ such that for every security parameter λ and every pair $(\text{stmt}, \text{wit})$ with $|\text{stmt}|, |\text{wit}| \leq p(\lambda)$ it holds that*

$$(\text{stmt}, \text{wit}) \in \text{Rel} \iff V_{\text{Rel}}(\text{stmt}, \text{wit}) = 1.$$

The NP language associated with Rel is

$$\mathcal{L}_{\text{Rel}} := \{\text{stmt} \in \{0, 1\}^* \mid \exists \text{wit} \in \{0, 1\}^* \text{ s.t. } (\text{stmt}, \text{wit}) \in \text{Rel}\}.$$

We refer to $\text{stmt} \in \mathcal{L}_{\text{Rel}}$ as a true statement (for Rel) and to $\text{stmt} \notin \mathcal{L}_{\text{Rel}}$ as a false statement.

When the relation is clear from context, we sometimes write $\mathcal{L}$ instead of $\mathcal{L}_{\text{Rel}}$, and simply write $\text{Rel}(\text{stmt}, \text{wit}) = 1$ as shorthand for $(\text{stmt}, \text{wit}) \in \text{Rel}$.

Hard relations. Intuitively, a relation is *hard* if it is easy to generate valid statement-witness pairs, but computationally infeasible to find a witness given only the statement. We formalize this via a sampling algorithm.

Definition 3.8 (Hard relation). *Let Rel be a relation equipped with a probabilistic polynomial-time sampling algorithm $\text{Rel.RGen}(1^\lambda)$ that outputs pairs $(\text{stmt}, \text{wit})$ such that $(\text{stmt}, \text{wit}) \in \text{Rel}$. For a PPT adversary $\mathcal{A}$, define its relation-hardness advantage as*

$$\text{Adv}_{\text{Rel}, \mathcal{A}}^{\text{hard}}(\lambda) := \Pr \left[(\text{stmt}, \text{wit}') \in \text{Rel} \mid \begin{array}{l} (\text{stmt}, \text{wit}) \leftarrow \text{Rel.RGen}(1^\lambda) \\ \text{wit}' \leftarrow \mathcal{A}(\text{stmt}) \end{array} \right].$$

We say that Rel is hard if $\text{Adv}_{\text{Rel}, \mathcal{A}}^{\text{hard}}(\lambda)$ is negligible for every PPT adversary $\mathcal{A}$.

3.3 Adaptor Signatures

We recall here the standard notion of adaptor signatures.

Definition 3.9. An adaptor signature scheme $\text{AS}_{\Sigma, \text{Rel}}$ w.r.t. a signature scheme $\Sigma = (\text{KGen}, \text{Sign}, \text{Verify})$ and a hard relation Rel consists of a tuple of four algorithms $\text{AS}_{\Sigma, \text{Rel}} = (\text{pSign}, \text{pVf}, \text{Adapt}, \text{Extract})$ defined as:

- $\tilde{\sigma} \leftarrow \text{pSign}(\text{sk}, m, \text{stmt})$. The pre-signing algorithm is a PPT algorithm that on input a secret key sk , message $m \in \{0, 1\}^{l_m}$, and statement stmt , outputs a pre-signature $\tilde{\sigma}$.
- $b \leftarrow \text{pVf}(\text{vk}, m, \tilde{\sigma}, \text{stmt})$. The pre-verification algorithm is a DPT algorithm that on input a verification key vk , message $m \in \{0, 1\}^{l_m}$, pre-signature $\tilde{\sigma}$, and statement stmt , outputs a bit b .
- $\sigma \leftarrow \text{Adapt}(\text{vk}, \tilde{\sigma}, \text{wit})$. The adapting algorithm is a PPT algorithm that on input a verification key vk , pre-signature $\tilde{\sigma}$, and a witness wit for the statement stmt outputs an adapted signature σ .
- $\text{wit} =: \text{Extract}(\text{vk}, \tilde{\sigma}, \sigma, \text{stmt})$. The extracting algorithm is a DPT algorithm that on input a verification key vk , a pre-signature $\tilde{\sigma}$, a signature σ and a statement stmt , outputs a witness wit , or $\perp$.

We often omit the subscript from $\text{AS}_{\Sigma, \text{Rel}}$ writing AS to improve readability if the signature scheme Σ and the hard relation Rel are clear from the context to improve readability. The correctness definitions look as follows:

Definition 3.10 (Pre-signature correctness). *An adaptor signature AS satisfies pre-signature correctness, if for all $\lambda \in \mathbb{N}$ and $m \in \{0, 1\}^{l_m}$:*

$$\Pr \left[\begin{array}{l} \text{pVf}(\text{vk}, m, \tilde{\sigma}, \text{stmt}) = 1 \wedge \\ \Sigma.\text{Verify}(\text{vk}, m, \sigma) = 1 \wedge \\ (\text{stmt}, \text{wit}') \in \text{Rel} \end{array} \mid \begin{array}{l} (\text{sk}, \text{vk}) \leftarrow \Sigma.\text{KGen}(1^\lambda), \\ (\text{stmt}, \text{wit}) \leftarrow \text{RGen}_{\text{Rel}}(1^\lambda), \\ \tilde{\sigma} \leftarrow \text{pSign}(\text{sk}, m, \text{stmt}), \\ \sigma \leftarrow \text{Adapt}(\text{vk}, \tilde{\sigma}, \text{wit}), \\ \text{wit}' := \text{Extract}(\text{vk}, \tilde{\sigma}, \sigma, \text{stmt}) \end{array} \right] = 1.$$

For security against a malicious seller we follow the *full extractability* notion proposed by Dai, Okamoto, and Yamamoto [16], which combines and extends the security properties of witness extractability and adaptor unforgeability from [24] to the multiple query setting.

Definition 3.11 (Extractability). *An adaptor signature scheme AS is extractable, if for every PPT adversary $\mathcal{A}$ there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$*

$$\Pr[\text{Ext}_{\mathcal{A}, \text{AS}}(\lambda) = 1] \leq \nu(\lambda) ,$$

where the experiment $\text{Ext}_{\mathcal{A}, \text{AS}}$ is described in Figure 3, and the probability is taken over the random choices of all probabilistic algorithms.

$\text{Ext}_{\mathcal{A}, \text{AS}}(\lambda)$	$\text{NewStmt}(\lambda)$
1 : $(\text{sk}, \text{vk}) \leftarrow \Sigma.\text{KGen}(1^\lambda); b \leftarrow 1; \mathcal{S}, \mathcal{T}, \mathcal{C} \leftarrow \emptyset$	1 : $(\text{stmt}, \text{wit}) \leftarrow \text{RGen}_{\text{Rel}}(1^\lambda)$
2 : $(m^*, \sigma^*) \leftarrow \mathcal{A}(\text{vk})^{\text{NewStmt}(\lambda), \text{Sign}(\text{sk}, \cdot), \text{pSign}(\text{sk}, \cdot, \cdot)}$	2 : $\mathcal{C} \leftarrow \mathcal{C} \cup \{\text{stmt}\}$
3 : assert $\Sigma.\text{Verify}(\text{vk}, m^*, \sigma^*)$	3 : return stmt
4 : assert $(m^* \notin \mathcal{S})$	
5 : for $(\text{stmt}, \tilde{\sigma}) \in \mathcal{T}[m^*]$ s.t. $\text{stmt} \notin \mathcal{C}$ do	<u>$\text{Sign}(\text{sk}, m)$</u>
6 : if $(\text{stmt}, \text{AS}.\text{Extract}(\text{vk}, \tilde{\sigma}, \sigma^*, \text{stmt})) \in \text{Rel}$ then	1 : $\sigma \leftarrow \Sigma.\text{Sign}(\text{sk}, m)$
7 : $b \leftarrow 0$	2 : $\mathcal{S} \leftarrow \mathcal{S} \cup \{m\}$
8 : return b	3 : return σ
	<u>$\text{pSign}(\text{sk}, m, \text{stmt})$</u>
	1 : $\tilde{\sigma} \leftarrow \text{AS}.\text{pSign}(\text{sk}, m, \text{stmt})$
	2 : $\mathcal{T}[m] \leftarrow \mathcal{T}[m] \cup \{(\text{stmt}, \tilde{\sigma})\}$
	3 : return $\tilde{\sigma}$

Figure 3: The security game $\text{Ext}_{\mathcal{A}, \text{AS}}(1^\lambda)$.

For security against a malicious buyer, we have the usual notion of *adaptability*.

Definition 3.12 (Pre-signature adaptability). *An adaptor signature scheme AS satisfies pre-signature adaptability if for all $\lambda \in \mathbb{N}$, messages $m \in \{0, 1\}^*$, valid statement/witness pairs $(\text{stmt}, \text{wit}) \in \text{Rel}$, verification keys vk and pre-signatures $\tilde{\sigma} \in \{0, 1\}^*$ we have that*

$$\text{pVf}(\text{vk}, m, \tilde{\sigma}, \text{stmt}) = 1 \text{ implies } \Sigma.\text{Verify}(\text{vk}, m, \text{Adapt}(\text{vk}, \tilde{\sigma}, \text{wit})) = 1.$$

We also introduce the notion of *strong canonical signing* which, on a high level, guarantees that it is computationally hard to distinguish a signature that is generated using pSign and Adapt from a signature that is generated using the signing procedure Sign of the underlying signature scheme. We introduce this notion following [11, 20]:

Definition 3.13 (Strong canonical signing). *Let $\text{AS}_{\Sigma, \text{Rel}}$ be an adaptor signature scheme with respect to a base signature scheme Σ and a relation Rel .*

For every valid statement-witness pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, define the canonical signing algorithm Sign^{can} by

$$\begin{aligned} &\text{Sign}^{\text{can}}(\text{sk}, \text{vk}, m, \text{stmt}, \text{wit}) : \\ &\quad \tilde{\sigma} \leftarrow \text{pSign}(\text{sk}, m, \text{stmt}), \quad \sigma \leftarrow \text{Adapt}(\text{vk}, \tilde{\sigma}, \text{wit}), \quad \text{return } \sigma. \end{aligned}$$

We say that AS has strong canonical signing with respect to Σ if for every $\lambda \in \mathbb{N}$, every key pair (sk, vk) in the support of $\Sigma.\text{KGen}(1^\lambda)$, every message m , and every valid statement-witness pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, the following two distributions are computationally indistinguishable:

$$\{(\text{vk}, m, \text{stmt}, \text{wit}, \text{Sign}(\text{sk}, m))\} \stackrel{\mathcal{E}}{\approx} \{(\text{vk}, m, \text{stmt}, \text{wit}, \text{Sign}^{\text{can}}(\text{sk}, \text{vk}, m, \text{stmt}, \text{wit}))\}.$$

We highlight that this is the strongest current notion when it comes to witness-confidentiality in the signature.

3.4 Pseudorandom Encryption

We formally introduce the algorithms for the encryption scheme below and define the pseudorandomness property.

Definition 3.14 (Private Key Encryption). *Let $(\mathcal{D}_K, \mathcal{D}_M, \mathcal{D}_C, \mathcal{D}_R)$ be the secret key space, message space, ciphertext space and randomness space, respectively. A private key encryption scheme $\text{SKE} = (\text{KGen}, \text{Enc}, \text{Dec})$ is defined as follows:*

- $k \leftarrow \text{KGen}(1^\lambda)$. The key generation algorithm takes as input the security parameter and generates a key k from the key space $\mathcal{D}_K$.
- $c \leftarrow \text{Enc}(k, m)$. The encryption algorithm takes as input a key $k \in \mathcal{D}_K$ and a message $m \in \mathcal{D}_M$, and returns a randomized ciphertext $c \in \mathcal{D}_C$. This means that Enc samples randomness $r \in \mathcal{D}_R$ and returns a deterministic $c = \text{Enc}(k, m; r)$.
- $m \leftarrow \text{Dec}(k, c)$. The decryption algorithm takes as input a key $k \in \mathcal{D}_K$ and a ciphertext $c \in \mathcal{D}_C$, and returns a plaintext $m \in \mathcal{D}_M$.

We call a pseudorandom encryption scheme perfectly correct if for any $\lambda \in \mathbb{N}$, any sk in the support of Setup and any message m , it holds that:

$$\Pr[\text{Dec}(\text{sk}, \text{Enc}(\text{sk}, m)) = m] = 1$$

over the randomness of Enc .

This is, for example, achieved by the private-key version of the ElGamal encryption scheme.

Furthermore, we define the security notion of pseudorandomness.

$\text{PR}_{b, \mathcal{A}, \text{SKE}}(\lambda)$	$\mathcal{O}_{b, k}(m)$
1 : $k \leftarrow \text{KGen}(1^\lambda)$	1 : $c \leftarrow \text{Enc}(k, m)$
2 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_{b, k}(\cdot)}(1^\lambda)$	2 : if $b = 1$: $c \leftarrow_{\$} \{0, 1\}^{ c }$
3 : return b'	3 : return c

Figure 4: The security game $\text{PR}_{\mathcal{A}, \text{SKE}}(\lambda)$.

Definition 3.15 (Ciphertext pseudorandomness). *The encryption scheme SKE is said to have pseudorandom ciphertexts if for every PPT adversary $\mathcal{A}$ there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$ we have that*

$$|\Pr[\text{PR}_{0, \mathcal{A}, \text{SKE}}(\lambda) = 1] - \Pr[\text{PR}_{1, \mathcal{A}, \text{SKE}}(\lambda) = 1]| \leq \nu(\lambda).$$

$\text{IND-CPA}_{b,\mathcal{A},\text{SKE}}(\lambda)$	$\mathcal{O}_{b,k}(m_0, m_1)$
1 : $k \leftarrow \text{KGen}(1^\lambda)$	1 : $c \leftarrow \text{Enc}(k, m_b)$
2 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_{b,k}(\cdot)}(1^\lambda)$	2 : return c
3 : return b'	

Figure 5: The security game $\text{IND-CPA}_{\mathcal{A},\text{SKE}}(\lambda)$.

We also introduce the weaker security notion of standard IND-CPA security.

Definition 3.16 (IND-CPA Security). *The encryption scheme SKE is said to satisfy IND-CPA security if for every PPT adversary $\mathcal{A}$ there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$ we have that*

$$|\Pr[\text{IND-CPA}_{0,\mathcal{A},\text{SKE}}(\lambda) = 1] - \Pr[\text{IND-CPA}_{1,\mathcal{A},\text{SKE}}(\lambda) = 1]| \leq \nu(\lambda) .$$

3.5 (Public key) Homomorphic encryption

A homomorphic encryption scheme is defined as follows:

Definition 3.17 (Homomorphic Encryption). *A homomorphic encryption scheme $\text{HE} = (\text{KGen}, \text{Enc}, \text{Eval}, \text{Dec})$ is defined as follows:*

- $(\text{pk}, \text{sk}) \leftarrow \text{KGen}(1^\lambda)$. *The key generation algorithm takes as input a security parameter λ and outputs a public key pk and a secret key sk .*
- $c \leftarrow \text{Enc}(\text{pk}, m)$. *The encryption algorithm takes as input the public key pk and a plaintext message m , and outputs a ciphertext c .*
- $c_f \leftarrow \text{Eval}(\text{pk}, f, c_1, \dots, c_n)$. *The evaluation algorithm takes as input the public key pk , a function f from a supported class $\mathcal{F}$, and ciphertexts $c_1, \dots, c_n$, and outputs an evaluated ciphertext c_f .*
- $m_f \leftarrow \text{Dec}(\text{sk}, c_f)$. *The decryption algorithm takes as input the secret key sk and the evaluated ciphertext c_f , and outputs the plaintext result m_f .*

The homomorphic encryption scheme HE is correct if for all security parameters λ , messages $m_1, \dots, m_n$ in the plaintext space, and functions $f \in \mathcal{F}$, it holds that:

$$\Pr \left[m_f \neq f(m_1, \dots, m_n) \mid \begin{array}{l} (\text{pk}, \text{sk}) \leftarrow \text{KGen}(1^\lambda), \\ c_i \leftarrow \text{Encrypt}(\text{pk}, m_i), \\ c_f \leftarrow \text{Eval}(\text{pk}, f, c_1, \dots, c_n), \\ m_f \leftarrow \text{Decrypt}(\text{sk}, c_f) \end{array} \right] \leq \nu(\lambda).$$

Definition 3.18. *We say that a homomorphic encryption scheme is randomness homomorphic if for any honestly generated public key pk , any function $f \in \mathcal{F}$, any message m and any randomness r , it holds that*

$$\text{Eval}(\text{pk}, f, \text{Enc}(m; r)) = \text{Enc}(f(m); f(r)).$$

Definition 3.19. *We say that a homomorphic encryption scheme has fully re-randomizable ciphertexts if there is a PPT algorithm CtRand such that, given any honestly generated public key pk , any message m and any randomness r , the following two distributions*

$$\{\text{CtRand}(\text{pk}, \text{Enc}(m; r))\} \stackrel{s}{\approx} \{\text{Enc}(\text{pk}, m)\}$$

are statistically indistinguishable.

We also recall the usual notion of circuit privacy, in the following form.

Definition 3.20. *We say that a homomorphic encryption scheme is circuit private if for any given message m , function $f \in \mathcal{F}$ and honestly generated public key $\mathbf{pk}$, the following two distributions*

$$\{\text{Eval}(\mathbf{pk}, f, \text{Enc}(\mathbf{pk}, m))\} \stackrel{s}{\approx} \{\text{Enc}(\mathbf{pk}, f(m))\}$$

are statistically indistinguishable.

Notice that this property can be achieved by combining perfect correctness and the two properties above, i.e. ciphertext re-randomizability and being homomorphic in the randomness.

We also define a new notion of re-randomizable messages.

Definition 3.21. *We say that a homomorphic encryption scheme has fully re-randomizable messages if there is a PPT algorithm MsgRand such that, given any honestly generated public key $\mathbf{pk}$ and any non-zero message m , the following two distributions*

$$\{\text{MsgRand}(\mathbf{pk}, \text{Enc}(\mathbf{pk}, m))\} \stackrel{s}{\approx} \{\text{Enc}(\mathbf{pk}, m')\}$$

are statistically indistinguishable, where in the second distribution m' is uniformly sampled in the message space, conditioned on not being zero.

Moreover, we require that the following two distributions

$$\{\text{MsgRand}(\mathbf{pk}, \text{Enc}(\mathbf{pk}, 0; r))\} \stackrel{s}{\approx} \{\text{Enc}(\mathbf{pk}, 0)\}$$

are also statistically indistinguishable.

Remark 3.22 (Instantiation for the concrete scheme). *For our concrete construction, we only require evaluation for the class of quadratic functions, and this is provided by the BGN encryption scheme [6]. Recall that the setup is a group $\mathbb{G} = \langle g \rangle$ of order $n = pq$, with hidden p and q . The secret key is the factorization, and the public key is $h \in \langle g^p \rangle$. The encryption of a message m is simply $c = g^m \cdot h^r$, which is IND-CPA under the hidden subgroup assumption. Notice then that this scheme has fully re-randomizable ciphertexts, simply given by*

$$\text{CtRand}(h, c) = c \cdot h^r,$$

for a freshly sampled uniform $r \in \mathbb{Z}_n$. Moreover, honest key generation can be attested by adding a NIZK proof that the public key modulus n is a proper RSA modulus, for example using techniques from [17, 33].⁷ Fortunately, the BGN scheme satisfies the required perfect completeness to ensure that if a NIZK proof for a key exists, this key will behave correctly. It also realizes fully re-randomizable messages, by

$$\text{MsgRand}(h, c) = c^{r_1} \cdot h^{r_2},$$

for freshly sampled uniform $r_1, r_2 \in \mathbb{Z}_n$.

⁷We highlight that this NIZK proof is very simple and lightweight since it only needs to certify something about an RSA modulus. It is much more lightweight than the type of NIZK proof that would be required to make the naïve scheme (beginning of Section 2) work.

3.6 Sigma protocols

We recall the definition of a Σ -protocol [3].

Definition 3.23 (Σ -protocol). *A Sigma protocol for a relation Rel consists of four algorithms:*

Prove₁(stmt, wit): *Takes as input the statement stmt and the witness wit, and outputs a commitment com and a state st.*

Chal(stmt, com): *Takes as input the statement stmt and the commitment com, and outputs a public-coin challenge c.*

Prove₂(st, c): *Takes as input the state st and the challenge c, and outputs a proof π .*

Verify(com, π): *Takes as input the commitment com and the proof π , and outputs either 0 or 1.*

A Σ -protocol for relation Rel has the above syntax and satisfies the following properties:

Completeness: *if P and V follow the protocol, then V always accepts;*

κ -Special Soundness: *There exists a PPT extractor that, given stmt and κ accepting conversations $(\text{com}, c_1, \pi_1), \dots, (\text{com}, c_\kappa, \pi_\kappa)$, where $c_i \neq c_j$ for all $i, j \in [\kappa]$, efficiently computes wit with $(\text{wit}, \text{stmt}) \in \text{Rel}$;*

Honest-verifier zero-knowledge (HVZK): *There exists a PPT simulator Sim, which on input stmt and c, outputs a transcript (com, c, π) with the same distribution as a real conversation between the honest P and V on input stmt and where V chooses c as a challenge*

In this work, we slightly differ from the standard setting and also introduce a crs that all of the algorithms receive as an input. Instead of proving κ -special soundness, we proceed to directly proving knowledge soundness:

Definition 3.24 (Knowledge Soundness [12]). *A sigma protocol with a CRS is knowledge sound if there exists an extractor $\text{Ext} = (\text{Ext}_1, \text{Ext}_2)$ such that the following two quantities are negligible:*

$$|\Pr[\text{crs} \leftarrow \text{Setup}(1^\lambda); 1 \leftarrow \mathcal{A}(\text{crs})] - \Pr[(\text{crs}, \text{st}) \leftarrow \text{Ext}_1(1^\lambda); 1 \leftarrow \mathcal{A}(\text{crs})]|$$

and

$$\Pr[\text{crs}_{\text{Ext}} \leftarrow \text{Ext}_1(1^\lambda); (\text{stmt}, \pi) \leftarrow \mathcal{A}(\text{crs}_{\text{Ext}}) : \text{Verify}(\text{crs}_{\text{Ext}}, \text{stmt}, \pi) = 1 \wedge \text{Rel}(\text{stmt}, \text{Ext}_2(\text{st}, \text{stmt}, \pi)) = 0]$$

3.7 Non-interactive Zero-Knowledge Proofs

In this section, we define non-interactive zero-knowledge proofs [5, 32, 26]. We start with the syntax.

Definition 3.25 (Non-interactive zero-knowledge). *A NIZK for a relation Rel consists of the following three algorithms:*

Setup(1^λ): *Takes as input a unary representation of the security parameter λ and outputs the common reference string crs.*

Prove(crs, stmt, wit): *Takes as input the common reference string crs, a statement stmt and a witness wit, and outputs a proof π .*

$\text{Verify}(\text{crs}, \text{stmt}, \pi)$: Takes as input the common reference string crs , a statement stmt and a proof π , and outputs either 0 or 1.

A NIZK proof system is complete if, for all $\lambda \in \mathbb{N}$, and for all $\text{crs} \in \text{Supp}(\text{Setup})(1^\lambda)$ and all statement-witness pairs in the relation $(\text{stmt}, \text{wit}) \in \text{Rel}$

$$\Pr[\text{Verify}(\text{crs}, \text{stmt}, \text{Prove}(\text{crs}, \text{stmt}, \text{wit})) = 1] = 1$$

In addition to completeness, we also introduce the notions of soundness and zero-knowledge. Furthermore, a NIZK can also be knowledge sound, in which case it basically satisfies Definition 3.24.

Definition 3.26 (Soundness). Given a NIZK for a relation Rel and the corresponding language $\mathcal{L}$, we say that it is sound, if for all PPT $\mathcal{A}$, it holds that:

$$\Pr \left[\begin{array}{c} \text{Verify}(\text{crs}, \text{stmt}, \pi) = 1 \\ \wedge \text{stmt} \notin \mathcal{L} \end{array} \middle| \begin{array}{c} \text{crs} \leftarrow \text{Setup}(1^\lambda) \\ (\text{stmt}, \pi) \leftarrow \mathcal{A}(\text{crs}) \end{array} \right] \leq \nu(\lambda)$$

Definition 3.27 (Zero-Knowledge). Given a NIZK for a relation Rel , let $\text{Sim} = (\text{Sim}_1, \text{Sim}_2)$ be a pair of algorithms, where Sim_1 outputs a simulated CRS together with a simulation trapdoor and Sim_2 generates a proof from the trapdoor and the statement, without receiving a witness. Define $\text{Sim}'(\text{crs}, \text{st}, \text{stmt}, \text{wit}) = \text{Sim}_2(\text{crs}, \text{st}, \text{stmt})$ for $(\text{stmt}, \text{wit}) \in \text{Rel}$, and $\text{Sim}'(\text{crs}, \text{st}, \text{stmt}, \text{wit}) = \perp$ for $(\text{stmt}, \text{wit}) \notin \text{Rel}$. We say that NIZK is zero-knowledge if, for all PPT $\mathcal{A}$, it holds that:

$$\left| \Pr \left[b = 1 \middle| \begin{array}{c} \text{crs} \leftarrow \text{Setup}(1^\lambda) \\ b \leftarrow \mathcal{A}^{\text{Prove}(\text{crs}, \cdot, \cdot)}(\text{crs}) \end{array} \right] - \Pr \left[b = 1 \middle| \begin{array}{c} (\text{crs}, \text{st}) \leftarrow \text{Sim}_1(1^\lambda) \\ b \leftarrow \mathcal{A}^{\text{Sim}'(\text{crs}, \text{st}, \cdot, \cdot)}(\text{crs}) \end{array} \right] \right| \leq \nu(\lambda)$$

4 Defining Adaptor Signatures with Advertisement

In this section, we show an extension of the syntax and properties of a standard adaptor signature for the setting where the seller will generate a (one-time) public advertisement adv committing to its witness. In terms of security, we borrow definitions from [16], but introduce two novel security properties: (a) *statement truth privacy* (Definition 4.6) and (b) *seller fairness* (Definition 4.5), a weakening of the security property of adaptability. We also recall the syntax and security games of standard adaptor signatures in Section 3.3.

Definition 4.1 (Adaptor signature with advertisement). An adaptor signature scheme with public advertisement $\text{AS}^{\text{Ad}}_{\Sigma, \text{Rel}}$ w.r.t. a signature scheme $\Sigma = (\text{KGen}, \text{Sign}, \text{Vf})$ and a hard relation Rel consists of a tuple of six algorithms $\text{AS}^{\text{Ad}}_{\Sigma, \text{Rel}} = (\text{Setup}, \text{AdGen}, \text{pSign}, \text{pVf}, \text{Adapt}, \text{Extract})$ defined as:

- $\text{pp} \leftarrow \text{Setup}(1^\lambda)$. The setup algorithm is a PPT algorithm that on input the security parameter 1^λ , outputs public parameters pp . All the following algorithms implicitly take the same pp as input.
- $(\text{adv}, \text{st}_S) \leftarrow \text{AdGen}(\text{stmt}, \text{wit})$. The advertisement generation algorithm is a PPT algorithm that on input a statement stmt and a witness wit , outputs an advertisement adv and a private seller state st_S .
- $(\tilde{\sigma}, \text{st}_B) \leftarrow \text{pSign}(\text{sk}, m, \text{adv}, \text{stmt})$. The pre-signing algorithm is a PPT algorithm that on input a secret key sk , a message $m \in \{0, 1\}^{l_m}$, an advertisement adv and a statement stmt , outputs a pre-signature $\tilde{\sigma}$ and private state information st_B .

- $b \leftarrow \text{pVf}(\text{vk}, m, \tilde{\sigma}, \text{adv}, \text{stmt})$. The pre-verification algorithm is a DPT algorithm that on input a verification key vk , message $m \in \{0, 1\}^{l_m}$, pre-signature $\tilde{\sigma}$, an advertisement adv and a statement stmt , outputs a bit b .
- $\sigma/\perp \leftarrow \text{Adapt}(\text{vk}, m, \tilde{\sigma}, \text{st}_S, \text{wit})$. The adapting algorithm is a PPT algorithm that on input a verification key vk , a message m , a pre-signature $\tilde{\sigma}$, the seller private state st_S and a witness wit , outputs an adapted signature σ , or $\perp$.
- $\text{wit}'/\perp =: \text{Extract}(\text{st}_B, \sigma)$. The extracting algorithm is a DPT algorithm that on input state information st_B and a signature σ , outputs a witness wit' or $\perp$.

In the definition above, notice that the extraction algorithm takes an additional private input in st_B compared to standard extraction. We therefore refer to this extension as adaptor signature with private extraction.

Next, we state correctness. Note that we only require extraction to recover some witness wit' for stmt ; it is not necessary to recover the particular witness wit used by AdGen .

Definition 4.2 (Correctness). An adaptor signature with advertisement $\text{AS}^{\text{Ad}}_{\Sigma, \text{Rel}}$ satisfies correctness, if for all $\lambda \in \mathbb{N}$ and $m \in \{0, 1\}^{l_m}$:

$$\Pr \left[\begin{array}{l} \text{pVf}(\text{vk}, m, \tilde{\sigma}, \\ \text{adv}, \text{stmt}) = 1 \\ \wedge \Sigma.\text{Vf}(\text{vk}, m, \sigma) = 1 \\ \wedge (\text{stmt}, \text{wit}') \in \text{Rel} \end{array} \middle| \begin{array}{l} (\text{sk}, \text{vk}) \leftarrow \Sigma.\text{KGen}(1^\lambda), \\ (\text{stmt}, \text{wit}) \leftarrow \text{Rel}.\text{RGen}(1^\lambda), \\ \text{pp} \leftarrow \text{Setup}(1^\lambda) \\ (\text{adv}, \text{st}_S) \leftarrow \text{AdGen}(\text{stmt}, \text{wit}), \\ (\tilde{\sigma}, \text{st}_B) \leftarrow \text{pSign}(\text{sk}, m, \text{adv}, \text{stmt}), \\ \sigma \leftarrow \text{Adapt}(\text{vk}, m, \tilde{\sigma}, \text{st}_S, \text{wit}), \\ \text{wit}' =: \text{Extract}(\text{st}_B, \sigma) \end{array} \right] = 1.$$

Security against Malicious Sellers. The usual notion of *extractability* of Dai, Okamoto, and Yamamoto [16] prevents a malicious seller from computing valid signatures that do not extract to a valid witness. We extend their definition to our setting with advertisement, preventing malicious sellers from possibly generating malformed advertisements that could help them in producing non-extractable signatures. Observe that in the classical notion of adaptor signatures, the AdGen algorithm outputs $(\text{stmt}, \emptyset)$, i.e., it is only advertising for the statement and does not additionally bind to a specific witness wit .

Definition 4.3 (Extractability). An adaptor signature with advertisement $\text{AS}^{\text{Ad}}_{\Sigma, \text{Rel}}$ is extractable, if for every PPT adversary $\mathcal{A}$ there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$

$$\Pr[\text{ExtAd}_{\mathcal{A}, \text{AS}^{\text{Ad}}}(\lambda) = 1] \leq \nu(\lambda),$$

where the experiment $\text{ExtAd}_{\mathcal{A}, \text{AS}^{\text{Ad}}}$ is described in Figure 6, and the probability is taken over the random coins of all algorithms.

Security against Malicious Buyers. A malicious buyer may submit malformed pre-signatures in an attempt to learn information about the seller's witness. There are two natural ways to protect the seller. The first is to require a pre-verification algorithm pVf . If pVf accepts, pre-signature adaptability guarantees the adaptability of the pre-signature into a valid signature. The second approach, without requiring a specific pVf procedure, is to require that failed adaptation attempts leak no information about the witness. We formalize this weaker guarantee as *seller fairness*. Informally, until the buyer submits a pre-signature that can be adapted into a valid signature, its view can be simulated without the witness.

$\text{ExtAd}_{\mathcal{A}, \text{AS}^{\text{Ad}}}(\lambda)$	$\text{NewStmt}(\lambda)$
1 : $(\text{sk}, \text{vk}) \leftarrow \Sigma.\text{KGen}(1^\lambda); \text{pp} \leftarrow \text{Setup}(1^\lambda)$ 2 : $b \leftarrow 1; \mathcal{S}, \mathcal{T}, \mathcal{X} \leftarrow \emptyset$ 3 : $(m^*, \sigma^*) \leftarrow \mathcal{A}(\text{pp}, \text{vk})^{\text{NewStmt}(\lambda), \text{Sign}(\text{sk}, \cdot), \text{pSign}(\text{sk}, \cdot, \cdot)}$ 4 : $\text{assert } \Sigma.\text{Vf}(\text{vk}, m^*, \sigma^*)$ 5 : $\text{assert } (m^* \notin \mathcal{S})$ 6 : for $(\text{adv}, \text{stmt}, \tilde{\sigma}, \text{st}_B) \in \mathcal{T}[m^*]$ s.t. $\text{stmt} \notin \mathcal{X}$ do 7 : if $(\text{stmt}, \text{Extract}(\text{st}_B, \sigma^*)) \in \text{Rel}$ then $b \leftarrow 0$ 8 : return b	1 : $(\text{stmt}, \text{wit}) \leftarrow \text{Rel}.\text{RGen}(1^\lambda)$ 2 : $(\text{adv}, \text{st}) \leftarrow \text{AS}^{\text{Ad}}.\text{AdGen}(\text{stmt}, \text{wit})$ 3 : $\mathcal{X} \leftarrow \mathcal{X} \cup \{\text{stmt}\}$ 4 : return $(\text{stmt}, \text{adv})$
$\text{pSign}(\text{sk}, m, \text{adv}, \text{stmt})$	$\text{Sign}(\text{sk}, m)$
1 : $(\tilde{\sigma}, \text{st}_B) \leftarrow \text{AS}^{\text{Ad}}.\text{pSign}(\text{sk}, m, \text{adv}, \text{stmt})$ 2 : if $\tilde{\sigma} = \perp$ then return $\perp$ 3 : $\mathcal{T}[m] \leftarrow \mathcal{T}[m] \cup \{(\text{adv}, \text{stmt}, \tilde{\sigma}, \text{st}_B)\}$ 4 : return $\tilde{\sigma}$	1 : $\sigma \leftarrow \Sigma.\text{Sign}(\text{sk}, m)$ 2 : $\mathcal{S} \leftarrow \mathcal{S} \cup \{m\}$ 3 : return σ

Figure 6: The security game $\text{ExtAd}_{\mathcal{A}, \text{AS}^{\text{Ad}}}(1^\lambda)$.

For completeness, we first extend pre-signature adaptability (Definition 3.12) to our setting. We then define our new notion of seller fairness in Definition 4.5.

Definition 4.4 (Pre-signature adaptability). *An adaptor signature scheme with advertisement AS^{Ad} satisfies pre-signature adaptability if there exists a negligible function ν , for any PPT adversary $\mathcal{A}$ and any satisfying statement-witness pair, i.e., $(\text{stmt}, \text{wit}) \in \text{Rel}$, it holds that*

$$\Pr \left[\begin{array}{c} \text{pVf}(\text{vk}, m, \tilde{\sigma}, \text{adv}, \text{stmt}) = 1 \\ \wedge \\ \Sigma.\text{Vf}(\text{vk}, m, \sigma) = 0 \end{array} \middle| \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda) \\ (\text{adv}, \text{st}_S) \leftarrow \text{AdGen}(\text{stmt}, \text{wit}) \\ (\text{vk}, m, \tilde{\sigma}) \leftarrow \mathcal{A}(\text{adv}, \text{stmt}) \\ \sigma \leftarrow \text{Adapt}(\text{vk}, m, \tilde{\sigma}, \text{st}_S, \text{wit}) \end{array} \right] \leq \nu(\lambda).$$

Seller fairness says that malformed pre-signatures do not reveal information about the witness merely because they fail to adapt. When a malicious buyer's pre-signature can be adapted, the game ends: the buyer learns the witness and the seller obtains the signature, matching an honest execution. To capture this intuition, we design an indistinguishability game where the adversary is asked to distinguish between a real game and a simulated game. In the real game, the adversary has oracle access to **Adapt**, where it can query with its choice of pre-signatures and see if the adapted signature verifies, without learning the signature. In other words, the oracle only reveals whether adaptation would yield a valid signature. This models the leakage available from failed adaptation attempts. The game halts immediately if an adapted signature verifies. The adversary is asked to return its state. If a query succeeds, the experiment suppresses the adversary's final output and returns $\perp$. In the simulated game, the adversary has oracle access to a simulated **Adapt**, where the simulator does not have access to the witness. The simulation ends when the adversary returns its

state. Intuitively, the definition captures the fact that, until an adapted pre-signature leads to a valid signature, no information about the witness is leaked.

Definition 4.5 (Seller fairness). *For any valid statement-witness pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, and any adversary PPT $\mathcal{A}$ there exists a PPT simulator $\text{Sim} = (\text{Sim}_1, \text{Sim}_2)$ such that the families $\{\text{FairReal}_{\mathcal{A}, \text{AS}^{\text{Ad}}}(\lambda, \text{stmt}, \text{wit})\}_{\lambda \in \mathbb{N}}$ and $\{\text{FairIdeal}_{\text{Sim}, \text{AS}^{\text{Ad}}}(\lambda, \text{stmt})\}_{\lambda \in \mathbb{N}}$ defined in Figure 7 are computationally indistinguishable.*

$\text{FairReal}_{\mathcal{A}, \text{AS}^{\text{Ad}}}(\lambda, \text{stmt}, \text{wit})$	$\mathcal{O}_P(\text{st}, \text{wit}, \text{vk}, \tilde{\sigma}, m)$
1 : $b \leftarrow 1$ // control flag	1 : if $b = 0$: then return $\perp$
2 : $(\text{adv}, \text{st}) \leftarrow \text{AdGen}(\text{stmt}, \text{wit})$	2 : $\sigma \leftarrow \text{Adapt}(\text{vk}, m, \tilde{\sigma}, \text{st}, \text{wit})$
3 : $z \leftarrow \mathcal{A}(1^\lambda, \text{stmt}, \text{adv})^{\mathcal{O}_P(\text{st}, \cdot)}$	3 : if $\Sigma.\text{Vf}(\text{vk}, m, \sigma)$ then
4 : if $b = 1$ then return z	4 : $b \leftarrow 0$;
5 : else return $\perp$	5 : return good;
$\text{FairIdeal}_{\text{Sim}, \text{AS}^{\text{Ad}}}(\lambda, \text{stmt})$	6 : return bad
1 : $b \leftarrow 1$ // control flag	
2 : $(\text{adv}, \text{st}) \leftarrow \text{Sim}_1(1^\lambda, \text{stmt})$	
3 : $z \leftarrow \mathcal{A}(1^\lambda, \text{stmt}, \text{adv})^{\text{Sim}_2(\text{st}, \cdot)}$	
4 : if $b = 1$ then return z	
5 : else return $\perp$	

Figure 7: The experiments FairReal and FairIdeal.

It is important to note that pre-signature adaptability protects the seller by ensuring that every pre-signature accepted by pVf can be adapted into a valid full signature. Seller fairness instead protects the seller in constructions without an additional pVf procedure; even if the buyer submits malformed pre-signatures, failed adaptation attempts do not leak any information about the witness.

We introduce a new property called *distributional statement truth privacy* which requires that the advertisement adv does not reveal if the statement stmt is true or false. To capture this, we first require that the distribution of true statements is computationally indistinguishable from the distribution of false statements. This condition is necessary, assuming that true and false statements are distinguishable before seeing adv , then no advertisement-generation procedure can provide truth privacy. More formally, let $D_{L_{\text{Rel}}}^T$ and $D_{L_{\text{Rel}}}^F$ be the distribution of statements that are in the language L_{Rel} and statements that are not in the language L_{Rel} , respectively. We want that, $\{\text{stmt}\}_{(\text{stmt}, \cdot) \leftarrow \mathcal{D}_{L_{\text{Rel}}}^T} \approx_c \{\text{stmt}\}_{\text{stmt} \leftarrow \mathcal{D}_{L_{\text{Rel}}}^F}$. The definition of truth privacy is distributional: it does not hide the truth value of an arbitrary adversarially chosen statement, but only preserves indistinguishability between prescribed true- and false-statement distributions.

Definition 4.6 (Distributional Statement truth privacy). *An adaptor signature scheme with advertisement AS^{Ad} satisfies statement truth privacy if there exists a PPT simulator $\mathcal{S}$ such that for every PPT distinguisher $\mathcal{D}$ there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$*

$$\left| \Pr \left[\mathcal{D}(1^\lambda, \text{stmt}, \text{adv}) = 1 \mid \begin{array}{l} (\text{stmt}, \text{wit}) \leftarrow \$ D_{L_{\text{Rel}}}^T, \\ \text{adv} \leftarrow \text{AdGen}(\text{stmt}, \text{wit}) \end{array} \right] - \Pr \left[\mathcal{D}(1^\lambda, \text{stmt}, \text{adv}) = 1 \mid \begin{array}{l} \text{stmt} \leftarrow \$ D_{L_{\text{Rel}}}^F, \\ \text{adv} \leftarrow \mathcal{S}(1^\lambda, \text{stmt}) \end{array} \right] \right| \leq \nu(\lambda)$$

Canonical signing. To guarantee confidentiality of the witness for generated signatures, we also adapt the notion of canonical signing to the setting with advertisement. It essentially follows the definition for the standard setting (Definition 3.13), but requires the guarantee to hold for every advertisement and corresponding private seller state in the support of honest advertisement generation. Formally,

Definition 4.7 (Strong canonical signing). *Let $\text{AS}_{\Sigma, \text{Rel}}^{\text{Ad}}$ be an adaptor signature scheme with advertisement for a base signature scheme Σ and a relation Rel . For every valid statement-witness pair $(\text{stmt}, \text{wit}) \in \text{Rel}$ and every pair $(\text{adv}, \text{st}_S)$ in the support of $\text{AdGen}(\text{stmt}, \text{wit})$, define the canonical signing algorithm Sign^{can} by*

$$\begin{aligned} & \text{Sign}^{\text{can}}(\text{sk}, \text{vk}, m, \text{stmt}, \text{wit}, \text{adv}, \text{st}_S) : \\ & \quad (\tilde{\sigma}, \text{st}_B) \leftarrow \text{pSign}(\text{sk}, m, \text{adv}, \text{stmt}), \\ & \quad \sigma \leftarrow \text{Adapt}(\text{vk}, m, \tilde{\sigma}, \text{st}_S, \text{wit}), \\ & \quad \text{return } \sigma. \end{aligned}$$

We say that AS^{Ad} has strong canonical signing with respect to Σ if for every $\lambda \in \mathbb{N}$, every key pair (sk, vk) in the support of $\Sigma.\text{KGen}(1^\lambda)$, every message m , and every valid statement-witness pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, and every pair $(\text{adv}, \text{st}_S)$ in the support of $\text{AdGen}(\text{stmt}, \text{wit})$, the following two distributions are computationally indistinguishable:

$$\begin{aligned} & \{(\text{vk}, m, \text{stmt}, \text{wit}, \text{adv}, \text{Sign}(\text{sk}, m))\} \\ & \stackrel{\mathcal{C}}{\approx} \{(\text{vk}, m, \text{stmt}, \text{wit}, \text{adv}, \text{Sign}^{\text{can}}(\text{sk}, \text{vk}, m, \text{stmt}, \text{wit}, \text{adv}, \text{st}_S))\}. \end{aligned}$$

5 From Discrete-logarithm to NP Adaptors for Schnorr (and More)

In this section, we investigate adaptor signatures with advertisement and study their feasibility for generic languages. Our main result is that we can turn a “basic” adaptor signature scheme AS for a specific language (for example, adaptor Schnorr signatures for the discrete logarithm relation) into one for generic languages while keeping the same fixed underlying signature scheme. In particular, we give a construction of adaptor signatures with advertisement for all of NP, using Schnorr’s signatures as the base signature scheme.

The main idea is to encrypt the witness twice: under an encryption scheme LE , for which we can sell secret keys via AS , and under a homomorphic encryption scheme HE , adding a proof π_{CE} that the two ciphertexts encrypt the same message. The buyer will then evaluate the verification of the NP relation under the hood of the HE using homomorphic evaluation, and use the result to mask the AS pre-signature, so that if the witness does not verify, the pre-signature is statistically hidden. An honest seller will be able to decrypt the pre-signature if and only if the witness is valid; using the

HE secret key, the seller can adapt it and obtain a full signature σ . Finally, the buyer extracts the secret key LE from the adapted signature and will use it to decrypt the advertised witness.

We use the following primitives:

- A private key encryption scheme SKE.
- A public-key linearly homomorphic encryption scheme LE that is also linear in the randomness.
- A public-key homomorphic encryption scheme HE that is also linear in the randomness and satisfies full rerandomization (Definition 3.21) for ciphertexts and messages.
- A NIZK proof system NIZK_{HE} for the language $\mathcal{L}_{\text{HE}}$, for which statements are HE public keys and witnesses are the corresponding secret keys, represented by the randomness used in key generation.
- An adaptor signature scheme AS for the LE key relation Rel_{LE} ; its statements are LE public keys, its witnesses are corresponding LE secret keys, and $\mathcal{L}_{\text{LE}}$ denotes the associated language.
- A NIZK proof system NIZK_{CE} for plaintext equality between HE and LE ciphertexts, together with a NIZK NIZK_{bit} proving that the HE plaintexts are bits.
- The signature scheme Σ underlying AS.
- A hash function $H_{\text{mask}} : \mathcal{M}_{\text{HE}} \rightarrow \mathbb{Z}_n \times \mathbb{Z}_n \times \mathbb{Z}_n$ that will be used to generate random coins for message and ciphertext re-randomisation.
- A hash function $H_{\text{key}} : \mathcal{M}_{\text{HE}} \rightarrow \mathcal{K}_{\text{SKE}}$ that will be used to generate a key for the SKE encryption scheme.

The adaptor signature with advertisement protocol follows below, in Figure 8. We encode every NP witness as a fixed-length bit vector. Accordingly, the ciphertexts c_{HE} and c_{LE} below denote componentwise encryptions of that vector, and we suppress the component index when no confusion can arise. We likewise use a public fixed-length encoding for the pre-signatures of AS. The main theorem that we prove in this section is the following.

Theorem 5.1. *Assume there exists an adaptor signature scheme $\text{AS}_{\Sigma, \text{Rel}_{\text{LE}}}$ for the LE key relation, using Σ as the underlying signature scheme. Suppose the existence of a homomorphic encryption scheme HE, and the existence of a proof system NIZK_{CE}. Then there exists an adaptor signature with advertisement scheme AS^{Ad} for any NP relation, whose base signature scheme is again Σ .*

In particular, we will prove correctness (Definition 4.2), extractability (Definition 4.3), seller fairness (Definition 4.5), strong canonical signing (Definition 4.7), and statement truth privacy (Definition 4.6).

Instantiations. For the instantiation, we rely on group-based primitives. The base signature scheme is the Schnorr signature scheme in a group $\mathbb{G} = \langle g \rangle$ of prime order p . We instantiate LE with exponential ElGamal in $\mathbb{G}$. The key relation Rel_{LE} is then the usual discrete-logarithm relation $\{(Y, y) : Y = g^y\}$, and $\mathcal{L}_{\text{LE}}$ is its statement language. AS is the adaptor Schnorr signature for Rel_{LE} . HE is the BGN encryption scheme, which can evaluate quadratic formulas and, at the expense of expanding the witness, any formula. As shown in Remark 3.22, it has all the properties that we need. NIZK_{CE} is the proof system described below.

5.1 Ciphertext Equality Proofs

Let LE and HE be two encryption schemes that are bi-homomorphic in message and randomness, i.e., for any $(\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}}) \leftarrow \text{LE.KGen}(1^\lambda)$, $(\text{pk}_{\text{HE}}, \text{sk}_{\text{HE}}) \leftarrow \text{HE.KGen}(1^\lambda)$, messages m, m' and randomness r, r' it holds that

$$\text{LE.Add}(\text{LE.Enc}(\text{pk}_{\text{LE}}, m; r), \text{LE.Enc}(\text{pk}_{\text{LE}}, m'; r')) = \text{LE.Enc}(\text{pk}_{\text{LE}}, m + m'; r + r')$$

Setup(1^λ) $\text{crs}_{\text{HE}} \leftarrow \text{NIZK}_{\text{HE}}.\text{Setup}(1^\lambda)$ $\text{crs}_{\text{CE}} \leftarrow \text{NIZK}_{\text{CE}}.\text{Setup}(1^\lambda)$ return $(\text{crs}_{\text{HE}}, \text{crs}_{\text{CE}})$ AdGen(stmt, wit) $\text{pk}_{\text{HE}}, \text{sk}_{\text{HE}} \leftarrow \text{HE}.\text{KGen}(1^\lambda; r_k)$ $\pi_{\text{HE}} \leftarrow \text{NIZK}_{\text{HE}}.\text{Prove}(\text{crs}_{\text{HE}}, \text{pk}_{\text{HE}}, (\text{sk}_{\text{HE}}, r_k))$ $\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}} \leftarrow \text{LE}.\text{KGen}(1^\lambda)$ $c_{\text{HE}} := \text{HE}.\text{Enc}(\text{pk}_{\text{HE}}, \text{wit}; r_{\text{HE}})$ $c_{\text{LE}} := \text{LE}.\text{Enc}(\text{pk}_{\text{LE}}, \text{wit}; r_{\text{LE}})$ $x = (\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}})$ $w_{\text{CE}} = (\text{wit}, r_{\text{HE}}, r_{\text{LE}})$ $\pi_{\text{CE}} \leftarrow \text{NIZK}_{\text{CE}}.\text{Prove}(\text{crs}_{\text{CE}}, x, w_{\text{CE}})$ encode $1 - \text{Rel}(\text{stmt}, \cdot)$ as F $c^* = \text{HE}.\text{Eval}(\text{pk}_{\text{HE}}, F, c_{\text{HE}})$ $\text{adv} = (x, \pi_{\text{CE}}, \pi_{\text{HE}})$ $\text{st} = (\text{sk}_{\text{LE}}, \text{pk}_{\text{LE}}, \text{sk}_{\text{HE}}, \text{pk}_{\text{HE}}, c^*)$ return (adv, st) Mask($\text{pk}_{\text{HE}}, c^*, s$) $(u_b, u_r, u_a) \leftarrow H_{\text{mask}}(s)$ $c_b = \text{HE}.\text{CtRand}(\text{pk}_{\text{HE}}, c^*; u_b)$ $c^\dagger = \text{HE}.\text{MsgRand}(\text{pk}_{\text{HE}}, c_b; u_r)$ $c_k = \text{HE}.\text{Enc}(\text{pk}_{\text{HE}}, s; 0)$ $c_a = \text{HE}.\text{HomAdd}(\text{pk}_{\text{HE}}, c_k, c^\dagger)$ $c_m = \text{HE}.\text{CtRand}(\text{pk}_{\text{HE}}, c_a; u_a)$ return c_m Extract(st, σ) $\text{vk}, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}, c_{\text{LE}} = \text{st}$ $\text{sk}_{\text{LE}} \leftarrow \text{AS}.\text{Extract}(\text{vk}, \tilde{\sigma}_{\text{LE}}, \sigma, \text{pk}_{\text{LE}})$ if $\text{sk}_{\text{LE}} = \perp$ then return $\perp$ $\text{wit} \leftarrow \text{LE}.\text{Dec}(\text{sk}_{\text{LE}}, c_{\text{LE}})$ return wit	pSign(sk, m, advt, stmt) $x, \pi_{\text{CE}}, \pi_{\text{HE}} = \text{adv}$ assert $\text{NIZK}_{\text{CE}}.\text{Vf}(\text{crs}_{\text{CE}}, x, \pi_{\text{CE}})$ $\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}} = x$ assert $\text{NIZK}_{\text{HE}}.\text{Vf}(\text{crs}_{\text{HE}}, \text{pk}_{\text{HE}}, \pi_{\text{HE}})$ encode $1 - \text{Rel}(\text{stmt}, \cdot)$ as F $c^* = \text{HE}.\text{Eval}(\text{pk}_{\text{HE}}, F, c_{\text{HE}})$ $s \leftarrow \$ \mathcal{M}_{\text{HE}}$ $c_m \leftarrow \text{Mask}(\text{pk}_{\text{HE}}, c^*, s)$ $\tilde{\sigma}_{\text{LE}} \leftarrow \text{AS}.\text{pSign}(\text{sk}, m, \text{pk}_{\text{LE}})$ $k_L \leftarrow H_{\text{key}}(s)$ $\text{ct} \leftarrow \text{SKE}.\text{Enc}(k_L, \tilde{\sigma}_{\text{LE}})$ $\tilde{\sigma} = (c_m, \text{ct})$ $\text{st} = (\text{vk}, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}, c_{\text{LE}})$ return $(\tilde{\sigma}, \text{st})$ Adapt(vk, m, $\tilde{\sigma}$, st, wit) $\text{sk}_{\text{LE}}, \text{pk}_{\text{LE}}, \text{sk}_{\text{HE}}, \text{pk}_{\text{HE}}, c^* = \text{st}$ $c_m, \text{ct} = \tilde{\sigma}$ $s = \text{HE}.\text{Dec}(\text{sk}_{\text{HE}}, c_m)$ $\tilde{c}_m \leftarrow \text{Mask}(\text{pk}_{\text{HE}}, c^*, s)$ if $\tilde{c}_m \neq c_m$ then return $\perp$ $k_L \leftarrow H_{\text{key}}(s)$ $\tilde{\sigma}_{\text{LE}} = \text{SKE}.\text{Dec}(k_L, \text{ct})$ if $\text{AS}.\text{pVf}(\text{vk}, m, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}) \neq 1$ then return $\perp$ $\sigma \leftarrow \text{AS}.\text{Adapt}(\text{vk}, \tilde{\sigma}_{\text{LE}}, \text{sk}_{\text{LE}})$ if $\text{AS}.\text{Vf}(\text{vk}, m, \sigma) \neq 1$ then return $\perp$ return σ
---	--

Figure 8: Generic language Schnorr Adaptor Signature scheme AS^{Ad} .

and the same holds for HE as well. Furthermore, we require that we can multiply the underlying message of a ciphertext, together with its randomness, using a scalar c , i.e.,

$$\text{LE.Mul}(c, \text{LE.Enc}(\text{pk}_{\text{LE}}, m; r)) = \text{LE.Enc}(\text{pk}_{\text{LE}}, c \cdot m; c \cdot r)$$

this holds for HE as well.

Then we can construct a zero-knowledge proof with partial extraction for the following language

$$\mathcal{L}_{\text{CE}} = \left\{ \left(\text{pk}_{\text{LE}}, \text{pk}_{\text{HE}}, \{c_{\text{LE},i}\}_{i \in [\ell]}, \{c_{\text{HE},i}\}_{i \in [\ell]}, \{m_i\}_{i \in [\ell]}, \{r_{\text{LE},i}\}_{i \in [\ell]}, \{r_{\text{HE},i}\}_{i \in [\ell]} \right) : m_i \in \{0, 1\}, \right. \\ \left. c_{\text{LE},i} = \text{LE.Enc}(\text{pk}_{\text{LE}}, m_i; r_{\text{LE},i}) \text{ and } c_{\text{HE},i} = \text{HE.Enc}(\text{pk}_{\text{HE}}, m_i; r_{\text{HE},i}) \text{ for all } i \in [\ell] \right\} \quad (1)$$

Remark 5.2 (Why we prove equality of bits). *The restriction $m_i \in \{0, 1\}$ is needed for soundness when the ciphertexts live in plaintext groups of different, possibly unknown, orders. The equality protocol publishes integer responses, but its verification equations hold in the respective plaintext groups. Without a common a priori bound on the plaintext, a malicious prover could exploit wraparound (or the Chinese remainder theorem) to make ciphertexts of different integer messages satisfy the equations. Proving that the HE plaintext is a bit, and bounding the shared integer response below every plaintext order, rules this out. The equality proof then transfers the bit value to the LE ciphertext and to the extractable ciphertext used in its proof. Separate bitness proofs for the LE and extractable ciphertexts are therefore unnecessary.*

Concretely, proving that a HE ciphertext encrypts a bit can be achieved using the GOS NIZK [34].

We show in Appendix A how to realize a Σ protocol for this language, borrowing techniques from dlog equality proofs such as [13]. Below, we describe how to turn this Sigma protocol into a full-fledged NIZK.

NIZK for $\mathcal{L}_{\text{CE}}$ in the ROM. We turn the Sigma protocol presented in Appendix A into a NIZK using the Fiat-Shamir transformation. Its soundness includes a straight-line, partial extractor that recovers the common plaintext, but not the encryption randomness. This is the only extraction property used by our compiler. Furthermore, when relying on the Fiat-Shamir transformation, we are also able to remove the extractable-encryption CRS.

In more detail, we derive its CRS and Fiat-Shamir challenge from two domain-separated random oracles H_{crs} and H_{fs} :

$$\text{crs}_{\text{CE}} := H_{\text{crs}}(\text{CE-CRS} \parallel pp) \quad \text{Chal} := H_{\text{fs}}(\text{CE-FS} \parallel pp \parallel \text{stmt}_{\text{CE}} \parallel \text{com}) ,$$

where pp can, for example, be derived from the blockchain or are some other public parameters.

To obtain partial extraction in this scenario, we sample a fresh LE keypair $(\text{pk}_{\text{crs}}, \text{sk}_{\text{crs}})$ and independently generate the bit-proof CRS crs_{bit} . We program the relevant answer of H_{crs} as the structured value $\text{crs}_{\text{CE}} = (\text{pk}_{\text{crs}}, \text{crs}_{\text{bit}})$ and retain sk_{crs} . This ensures that the extractor can recover the plaintext encrypted in the proof, while the bit-proof CRS has its honest distribution.

Taking this into account, we use the partial extraction property proved in Lemma A.2 and the usual zero-knowledge treatment from Definition 3.27 in the remainder of this work.

5.2 Proof-of-Concept Implementation

We implemented the construction from Figure 8 for Schnorr signatures.⁸ The artifact is written in Rust and covers the full protocol flow, including `AdGen`, `pSign`, `Adapt`, and `Extract`. In our instantiation, `LE` is exponential ElGamal over `secp256k1`, and the underlying adaptor scheme is the standard Schnorr adaptor for the discrete-logarithm relation. The homomorphic encryption scheme `HE` is composite-order BGN [6], implemented using pairings through `PBC`. The private-key encryption scheme `SKE` is authenticated symmetric encryption, used only to wrap the underlying adaptor pre-signature.

The implementation represents witnesses as bit strings. The default example uses the small bounded-integer relation $ab + de = z$: the witness contains binary encodings of a, b, d, e , together with helper values for ab and de , yielding two multiplication constraints and one linear constraint. However, the prototype supports arbitrary relations. For the evaluation function F , the buyer first samples a seed, derives random coefficients $\rho_1, \dots, \rho_t$ for each quadratic constraint $f_1, \dots, f_t$, and evaluates the batched polynomial

$$F(w) = \sum_{i=1}^t \rho_i f_i(w)$$

under BGN, similar to batched linear combination techniques from various proving systems [9]: if all constraints are satisfied then $F(w) = 0$, while a nonzero residual vector is mapped to zero only with negligible probability over the buyer’s random choice.

The masking step is implemented bitwise for the key k_L , since BGN decryption is efficient only for small plaintext spaces, making its cost linear in the bit length of k_L and independent of the relation, while homomorphic evaluation grows with the constraint count. On this example relation, a release build on an Apple M3 Max machine took 0.42s for `AdGen`, 0.70s for `pSign`, 0.34s for `Adapt`, and 2.3ms for `Extract`, for a single end-to-end run of 1.45s. The small relation is chosen deliberately, to separate the compiler’s cost from the relation’s. The signature verifications appearing in the applications of Section 1.1 carry orders of magnitude more constraints and lie well beyond the timings above. Whether a degree-2 scheme admitting a large plaintext space would remove the bitwise masking overhead is an interesting question we leave open.

5.3 Formal Analysis

In this section, we present the formal security analysis of our adaptor signature construction from Figure 8. We start with the proof of correctness, followed by the proofs of extractability, seller fairness, strong canonical signing, and statement truth privacy.

5.3.1 Correctness

Proposition 5.3. *The scheme AS^{Ad} in Figure 8 is correct.*

Proof. For proving correctness, we consider a valid witness `wit` for the statement `stmt`. The advertisement is composed of $\mathbf{pk}_{HE}, c_{HE}, \mathbf{pk}_{LE}, c_{LE}$: honestly computed public keys with known secret keys $\mathbf{sk}_{HE}, \mathbf{sk}_{LE}$, encryptions of the same witness, and a valid proof π_{CE} . The pre-sign algorithm will then evaluate the relation circuit homomorphically on the encrypted witness c_{HE} ; by correctness of `HE`, validity of `wit` and honest encryption, the result c^* will encrypt 0. Thus, also $c^\dagger$ will be an encryption

⁸<https://anonymous.4open.science/r/generic-adaptors-C4D6/README.md>

of 0, and c_m will result in an encryption of s by correctness and the re-randomization properties of HE.

The full pre-signature will be composed by the pre-signature $\tilde{\sigma}_{\text{LE}}$ for the $\mathcal{L}_{\text{LE}}$ language, and ct will be an encryption of that under k_L . Notice that we are deriving the randomness for the Mask algorithm from s through H_{mask} , and using the same s to derive the key k_L from H_{key} . The correctness of HE allows us to obtain the correct s ; thus, the seller can recover all the randomness and, since HE.Eval is deterministic, correctly recompute all the values $c^*, c^\dagger, c_k, c_m$. Finally, from the right $k_L = H_{\text{key}}(s)$, by correctness of SKE, the seller gets $\tilde{\sigma}_{\text{LE}}$ and using the adapting algorithm of AS it will get a valid signature σ , since sk_{LE} is a valid witness for pk_{LE} .

The correctness of extract follows from the correctness of the underlying adaptor signature AS for $\mathcal{L}_{\text{LE}}$. The buyer is able to extract sk_{LE} from σ , and then by correctness of LE it gets the original witness wit by decrypting c_{LE} . $\square$

5.3.2 Extractability

Proposition 5.4. *Let Rel be a hard relation and let AS be an extractable adaptor signature scheme for Rel_{LE} (Definition 3.11). Suppose that HE and LE are IND-CPA secure, that HE is randomness homomorphic and has fully rerandomizable ciphertexts and messages, and that SKE is IND-CPA secure. Assume that $\mathcal{M}_{\text{HE}}$ is a finite additive group and that $1/|\mathcal{M}_{\text{HE}}|$ is negligible in λ . Model H_{mask} and H_{key} as programmable random oracles, with H_{key} returning keys distributed as $k \leftarrow \text{SKE.KGen}(1^\lambda)$. Suppose further that NIZK_{CE} has the partial knowledge soundness of Lemma A.2 and is zero knowledge. Finally, suppose that NIZK_{HE} is sound and zero knowledge. Then the AS^{Ad} scheme in Figure 8 is extractable as in Definition 4.3.*

Proof. Let $\mathcal{A}$ be a PPT adversary in the extractability game $\text{ExtAd}_{\mathcal{A}, \text{AS}^{\text{Ad}}}$ for adaptor schemes with advertisement. We bound its advantage using relation hardness and the extractability of AS. Let G_0 be $\text{ExtAd}_{\mathcal{A}, \text{AS}^{\text{Ad}}}(\lambda)$, and let E_{forge} denote the event that the final output (m^*, σ^*) satisfies the two forgery conditions $\Sigma.\text{Vf}(\text{vk}, m^*, \sigma^*) = 1$ and $m^* \notin \mathcal{S}$, and that no record in $\mathcal{T}[m^*]$ whose statement lies outside $\mathcal{X}$ extracts a valid witness. Partition this event as

$$\begin{aligned} \text{E}_{\text{rel}} &:= \left\{ \omega \in \text{E}_{\text{forge}} \left| \begin{array}{l} \exists (\text{adv}, \text{stmt}, \tilde{\sigma}, \text{st}_B) \in \mathcal{T}[m^*] : \\ \text{stmt} \in \mathcal{X} \wedge (\text{stmt}, \text{AS}^{\text{Ad}}.\text{Extract}(\text{st}_B, \sigma^*)) \in \text{Rel} \end{array} \right. \right\}, \\ \text{E}_{\text{sig}} &:= \left\{ \omega \in \text{E}_{\text{forge}} \left| \begin{array}{l} \forall (\text{adv}, \text{stmt}, \tilde{\sigma}, \text{st}_B) \in \mathcal{T}[m^*] : \\ (\text{stmt}, \text{AS}^{\text{Ad}}.\text{Extract}(\text{st}_B, \sigma^*)) \notin \text{Rel} \end{array} \right. \right\}. \end{aligned}$$

The condition in E_{forge} rules out a successful extraction for statements outside $\mathcal{X}$, so these events are disjoint and exhaustive. Consequently,

$$\Pr[\text{E}_{\text{forge}}] = \Pr[\text{E}_{\text{rel}}] + \Pr[\text{E}_{\text{sig}}].$$

Define $G_{\text{rel},0}$ to run G_0 and output one precisely on E_{rel} . Likewise, define $G_{\text{sig},0}$ to run G_0 and output one precisely on E_{sig} . Thus

$$\Pr[\text{E}_{\text{rel}}] = \Pr[G_{\text{rel},0} = 1] \quad \text{and} \quad \Pr[\text{E}_{\text{sig}}] = \Pr[G_{\text{sig},0} = 1].$$

We prove in Lemma 5.5 that $\Pr[G_{\text{rel},0} = 1]$ is negligible by reducing to the hardness of Rel. We prove in Lemma 5.6 that $\Pr[G_{\text{sig},0} = 1]$ is negligible by reducing to the extractability of AS. Since E_{forge} is

the output event of G_0 , we obtain

$$\Pr[G_0 = 1] = \Pr[G_{\text{rel},0} = 1] + \Pr[G_{\text{sig},0} = 1] \leq \nu(\lambda),$$

which concludes the proof. $\square$

Lemma 5.5 (Relation-witness branch). *Let $q_Y \geq 1$ be a polynomial upper bound on the number of NewStmt queries. If Rel is hard, HE and LE are IND-CPA secure, and NIZK_{CE} and NIZK_{HE} are zero knowledge, then there exists a PPT relation-hardness adversary $\mathcal{B}_{\text{rel}}$ such that*

$$\Pr[G_{\text{rel},0} = 1] \leq q_Y \text{Adv}_{\text{Rel}, \mathcal{B}_{\text{rel}}}^{\text{hard}}(\lambda) + \nu(\lambda).$$

In particular, $\Pr[G_{\text{rel},0} = 1]$ is negligible.

Proof. Consider the following games.

Game $G_{\text{rel},0}$. Run G_0 and output one exactly on $\mathbf{E}_{\text{rel}}$, as defined in the proof of Proposition 5.4.

Game $G_{\text{rel},1}$. Choose $J \leftarrow_{\$} [q_Y]$ at the beginning of the game. Write $(\text{stmt}_i, \text{wit}_i)$ for the relation pair sampled to answer the i -th NewStmt query. If $\mathbf{E}_{\text{rel}}$ occurs, let i^* be the least index for which there is a record $(\text{adv}_t, \text{stmt}_i, \tilde{\sigma}, \text{st}_B) \in \mathcal{T}[m^*]$ satisfying

$$(\text{stmt}_i, \text{AS}^{\text{Ad}}.\text{Extract}(\text{st}_B, \sigma^*)) \in \text{Rel}.$$

Output one exactly when $\mathbf{E}_{\text{rel}}$ occurs and $J = i^*$. This convention also removes any ambiguity if two NewStmt queries return the same statement. Since J is independent and uniform,

$$\Pr[G_{\text{rel},0} = 1] \leq q_Y \Pr[G_{\text{rel},1} = 1].$$

Game $G_{\text{rel},2}$. Generate the NIZK_{CE} and NIZK_{HE} CRSs with their simulation setups, and simulate both proofs in every advertisement returned by NewStmt. Zero knowledge of the two proof systems, hybridized over the polynomially many NewStmt queries, gives

$$|\Pr[G_{\text{rel},1} = 1] - \Pr[G_{\text{rel},2} = 1]| \leq \nu(\lambda).$$

Game $G_{\text{rel},3}$. In the advertisement returned by the J -th NewStmt query, replace the componentwise ciphertexts

$$(\text{HE}.\text{Enc}(\text{pk}_{\text{HE}}, \text{wit}_J), \text{LE}.\text{Enc}(\text{pk}_{\text{LE}}, \text{wit}_J))$$

by

$$(\text{HE}.\text{Enc}(\text{pk}_{\text{HE}}, 0^\ell), \text{LE}.\text{Enc}(\text{pk}_{\text{LE}}, 0^\ell)),$$

and simulate a new equality proof for the modified ciphertexts; the simulated NIZK_{HE} proof is unchanged. A hybrid over the ℓ components and the two encryption schemes reduces this transition to IND-CPA security of HE and LE. The reduction can answer every pSign query because the compiler's pre-signing algorithm is a public efficient transformation of the advertised ciphertexts and the reduction generates the signing key. Hence

$$|\Pr[G_{\text{rel},2} = 1] - \Pr[G_{\text{rel},3} = 1]| \leq \nu(\lambda).$$

To bound $G_{\text{rel},3}$, construct a relation-hardness adversary $\mathcal{B}_{\text{rel}}$. Its challenger samples

$$(\text{stmt}^\dagger, \text{wit}^\dagger) \leftarrow \text{Rel.RGen}(1^\lambda)$$

and gives only $\text{stmt}^\dagger$ to $\mathcal{B}_{\text{rel}}$. The reduction chooses $J \leftarrow \$ [q_Y]$, generates the signature key pair and the two simulated NIZK CRSs, and runs $\mathcal{A}$. It answers every **NewStmt** query other than the J -th one by sampling an honest relation pair and generating the corresponding ciphertexts with simulated proofs. For the J -th query it uses $\text{stmt}^\dagger$, encrypts 0^ℓ under fresh HE and LE public keys, and simulates both proofs. All signing and pre-signing queries are answered exactly as in $G_{\text{rel},3}$.

On receiving (m^*, σ^*) , the reduction examines the records in $\mathcal{T}[m^*]$ whose statement equals $\text{stmt}^\dagger$. For each such record it computes

$$\text{wit}' = \text{AS}^{\text{Ad}}.\text{Extract}(\text{st}_B, \sigma^*)$$

and returns the first wit' satisfying $(\text{stmt}^\dagger, \text{wit}') \in \text{Rel}$. Whenever $G_{\text{rel},3}$ outputs one, the definition of i^* guarantees that this search succeeds. Therefore

$$\Pr[G_{\text{rel},3} = 1] \leq \text{Adv}_{\text{Rel}, \mathcal{B}_{\text{rel}}}^{\text{hard}}(\lambda).$$

Combining the guessing bound with the two hybrid transitions proves the lemma. $\square$

Lemma 5.6 (Signature branch). *Under the assumptions of Proposition 5.4, for every PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}_{\text{sig}}$ such that*

$$\Pr[G_{\text{sig},0} = 1] \leq \Pr[\text{Ext}_{\mathcal{B}_{\text{sig}}, \text{AS}}(\lambda) = 1] + \nu(\lambda).$$

In particular, $\Pr[G_{\text{sig},0} = 1]$ is negligible.

Proof. Let q_P bound the total number of **pSign** queries, let q_H bound the total number of adversarial queries to H_{mask} and H_{key} , and let q_{fs} bound the total number of queries to the Fiat–Shamir oracle of NIZK_{CE} , including verification queries. Let L_{pre} be the public encoding length of an underlying pre-signature. The games in this branch are as follows.

Game $G_{\text{sig},0}$. Run G_0 and output one exactly on E_{sig} .

Game $G_{\text{sig},1}$. Let **BadKeys** be the event that an advertisement accepted by the **pSign** oracle contains $\text{pk}_{\text{HE}} \notin \mathcal{L}_{\text{HE}}$. Output zero if **BadKeys** occurs.

Game $G_{\text{sig},2}$. Generate the NIZK_{CE} CRS as $(\text{crs}_{\text{CE}}, \tau_{\text{CE}}) \leftarrow \text{NIZK}_{\text{CE}}.\text{Ext}_1(1^\lambda)$ and retain the extraction trapdoor τ_{CE} . In the random-oracle instantiation, program the corresponding H_{crs} value to crs_{CE} . Concretely, the extractable-encryption component of crs_{CE} is a fresh LE public key pk_{crs} , and $\tau_{\text{CE}} = \text{sk}_{\text{crs}}$.

Game $G_{\text{sig},3}$. For every advertisement accepted by the **pSign** oracle, compute

$$\widehat{\text{wit}} \leftarrow \text{NIZK}_{\text{CE}}.\text{Ext}_2(\tau_{\text{CE}}, x, \pi_{\text{CE}})$$

and output zero if $\widehat{\text{wit}} = \perp$. Let **BadExt** be the event that $\widehat{\text{wit}} = \perp$ or that the ciphertexts c_{HE} and c_{LE} do not both have openings to the bit vector $\widehat{\text{wit}}$.

Game $G_{\text{sig},4}$. On each **pSign** query, sample $s \leftarrow \$ \mathcal{M}_{\text{HE}}$. If s has already appeared in either random-oracle table or was used by an earlier **pSign** query, output zero and set **BadProg**. Otherwise sample

$$(u_b, u_r, u_a) \leftarrow \$ \mathbb{Z}_n^3, \quad k_L \leftarrow \text{SKE.KGen}(1^\lambda),$$

and program $H_{\text{mask}}(s) = (u_b, u_r, u_a)$ and $H_{\text{key}}(s) = k_L$ before constructing the pre-signature.

Game $G_{\text{sig},5}$. For every **pSign** query with $(\text{stmt}, \widehat{\text{wit}}) \notin \text{Rel}$, let $t = F(\widehat{\text{wit}}) = 1 - \text{Rel}(\text{stmt}, \widehat{\text{wit}}) \neq 0$ and replace the internal ciphertext c_b by $\text{HE.Enc}(\text{pk}_{\text{HE}}, t)$.

Game $G_{\text{sig},6}$. For every such query, replace $c^\dagger$ by $\text{HE.Enc}(\text{pk}_{\text{HE}}, s'; u')$, where $s' \leftarrow \$ \mathcal{M}_{\text{HE}} \setminus \{0\}$ and u' is fresh encryption randomness.

Game $G_{\text{sig},7}$. For every such query, replace c_a by $\text{HE.Enc}(\text{pk}_{\text{HE}}, s' + s; u')$.

Game $G_{\text{sig},8}$. For every such query, replace c_m by $\text{HE.Enc}(\text{pk}_{\text{HE}}, s'')$ for an independent $s'' \leftarrow \$ \mathcal{M}_{\text{HE}}$.

Game $G_{\text{sig},9}$. Set **BadReveal** and output zero if the adversary queries either $H_{\text{mask}}(s)$ or $H_{\text{key}}(s)$ for an s used in a **pSign** query satisfying $(\text{stmt}, \widehat{\text{wit}}) \notin \text{Rel}$.

Game $G_{\text{sig},10}$. For every such query, replace $\text{ct} = \text{SKE.Enc}(k_L, \tilde{\sigma}_{\text{LE}})$ by $\text{ct} = \text{SKE.Enc}(k_L, 0^{L_{\text{pre}}})$.

Soundness of NIZK_{HE} , followed by a union bound over the accepted **pSign** queries, gives $\Pr[\text{BadKeys}] \leq \nu(\lambda)$ and hence

$$|\Pr[G_{\text{sig},0} = 1] - \Pr[G_{\text{sig},1} = 1]| \leq \nu(\lambda).$$

The CRS output by $\text{NIZK}_{\text{CE}}.\text{Ext}_1$ is indistinguishable from an honest CRS, so $G_{\text{sig},1} \stackrel{\mathcal{C}}{\approx} G_{\text{sig},2}$. For the concrete equality proof the two CRS distributions are identical, as shown in Lemma A.2.

Partial knowledge soundness of NIZK_{CE} , including the standard union bound over Fiat–Shamir trials, gives the explicit bound

$$\Pr[\text{BadExt}] \leq q_P \ell \epsilon_{\text{bit}} + \frac{2\ell q_{\text{fs}}}{C} \leq \nu(\lambda).$$

The extractor's output is internal to the game, and the only change from $G_{\text{sig},2}$ to $G_{\text{sig},3}$ is the output-zero rule when extraction returns $\perp$. Thus

$$|\Pr[G_{\text{sig},2} = 1] - \Pr[G_{\text{sig},3} = 1]| \leq \Pr[\text{BadExt}].$$

Outside **BadExt**, every accepted **pSign** query has a well-defined extracted bit vector $\widehat{\text{wit}}$, and both advertised ciphertexts have openings to $\widehat{\text{wit}}$.

The value s sampled for each **pSign** query is uniform and independent of the preceding view. A union bound therefore gives

$$\Pr[\text{BadProg}] \leq \frac{q_P(q_H + q_P)}{|\mathcal{M}_{\text{HE}}|} \leq \nu(\lambda),$$

and hence $G_{\text{sig},3} \stackrel{\mathcal{C}}{\approx} G_{\text{sig},4}$.

For each $i \in \{4, \dots, 8\}$, let $\overline{G}_{\text{sig},i}$ be $G_{\text{sig},i}$ with a distinguished abort as soon as **BadReveal** occurs. We first compare these aborting games. Before the abort, neither programmed random-oracle value

at the hidden s has been revealed, so the rerandomization assumptions apply without exposing their coins.

Fix an invalid-witness query under this conditioning. Randomness homomorphism gives an opening of $c^* = \text{HE.Eval}(\text{pk}_{\text{HE}}, F, c_{\text{HE}})$ to $t = F(\widehat{\text{wit}}) \neq 0$. Full ciphertext rerandomizability therefore justifies the replacement of $c_b = \text{HE.CtRand}(\text{pk}_{\text{HE}}, c^*; u_b)$ by a fresh encryption of t . A hybrid over the at most q_P queries gives $\overline{G}_{\text{sig},4} \approx \overline{G}_{\text{sig},5}$.

Full message rerandomizability similarly gives $\overline{G}_{\text{sig},5} \approx \overline{G}_{\text{sig},6}$. Randomness homomorphism and the zero randomness of $c_k = \text{HE.Enc}(\text{pk}_{\text{HE}}, s; 0)$ give the exact identity

$$\text{HE.HomAdd}(\text{pk}_{\text{HE}}, c_k, c^\dagger) = \text{HE.Enc}(\text{pk}_{\text{HE}}, s + s'; u'),$$

so $\overline{G}_{\text{sig},6} \equiv \overline{G}_{\text{sig},7}$. Full ciphertext rerandomizability replaces the resulting c_m by a fresh encryption of $s + s'$. For every fixed s , the value $s + s'$ is uniform on $\mathcal{M}_{\text{HE}} \setminus \{s\}$ and therefore has statistical distance $1/|\mathcal{M}_{\text{HE}}|$ from uniform on the additive message group. Hence $\overline{G}_{\text{sig},7} \approx \overline{G}_{\text{sig},8}$.

In $G_{\text{sig},8}$, the public response to an invalid-witness query is independent of its hidden value s , except through the two random-oracle entries programmed at s . Both entries are uniform and hidden until s is queried. Consequently,

$$\Pr[\text{BadReveal}] \leq \frac{q_P q_H}{|\mathcal{M}_{\text{HE}}| - q_H - q_P} \leq \nu(\lambda).$$

The abort symbol is part of the output of each $\overline{G}_{\text{sig},i}$. The comparisons above therefore transfer the negligible bound on **BadReveal** from $\overline{G}_{\text{sig},8}$ to all preceding aborting games. Each $G_{\text{sig},i}$ is consequently indistinguishable from its aborting counterpart, which establishes the corresponding transitions between $G_{\text{sig},4}, \dots, G_{\text{sig},8}$. The same bound gives $G_{\text{sig},8} \approx G_{\text{sig},9}$. Conditioned on $\neg \text{BadReveal}$, every k_L used for an invalid-witness query is an independent, hidden key with the distribution of **SKE.KGen**. A hybrid over the at most q_P ciphertexts and IND-CPA security of **SKE** therefore gives $G_{\text{sig},9} \approx G_{\text{sig},10}$. Combining the game transitions,

$$\Pr[G_{\text{sig},0} = 1] \leq \Pr[G_{\text{sig},10} = 1] + \nu(\lambda).$$

We now construct $\mathcal{B}_{\text{sig}}$ in the extractability experiment for **AS**. It receives vk , has access to **Ext.NewStmt**, **Ext.Sign**, and **Ext.pSign**, and generates the public parameters of $G_{\text{sig},10}$, including $(\text{crs}_{\text{CE}}, \tau_{\text{CE}}) \leftarrow \text{NIZK}_{\text{CE}}.\text{Ext}_1(1^\lambda)$. Denote the underlying experiment's tables by $\mathcal{T}_u$ and $\mathcal{C}_u$. It simulates the programmed random oracles and the detectable abort rules **BadProg** and **BadReveal**. The reduction runs $\mathcal{A}(\text{pp}, \text{vk})$ and answers its oracle queries as follows.

- On a **NewStmt** query, call **Ext.NewStmt** to obtain pk_{LE} . Sample $(\text{stmt}, \text{wit}) \leftarrow \text{Rel.RGen}(1^\lambda)$ and a fresh HE key pair, encrypt wit under pk_{HE} and pk_{LE} , and generate π_{CE} and π_{HE} . Return $(\text{stmt}, \text{adv})$ and add stmt to $\mathcal{X}$.
- On a **Sign** query, forward the message to **Ext.Sign**, return the resulting signature, and add the message to $\mathcal{S}$.
- On a **pSign** query, verify and parse the advertisement, and return $\perp$ if the compiler's verification checks fail. Otherwise compute $\widehat{\text{wit}} \leftarrow \text{NIZK}_{\text{CE}}.\text{Ext}_2(\tau_{\text{CE}}, x, \pi_{\text{CE}})$. If $(\text{stmt}, \widehat{\text{wit}}) \notin \text{Rel}$, generate the public pre-signature as in $G_{\text{sig},10}$, using $0^{L_{\text{pre}}}$ and a dummy private state, without calling the underlying pre-signing oracle. If $(\text{stmt}, \widehat{\text{wit}}) \in \text{Rel}$, call **Ext.pSign**(m, pk_{LE}) to obtain $\tilde{\sigma}_{\text{LE}}$ and complete the compiler's pre-signature computation. In this case store $(\text{vk}, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}, c_{\text{LE}})$ as the buyer state.

When $\mathcal{A}$ returns (m^*, σ^*) , the reduction returns the same pair to its challenger. Outside **BadKeys** and **BadExt**, the view of $\mathcal{A}$ is distributed as in $G_{\text{sig},10}$. Buyer states for invalid-witness queries are hidden from $\mathcal{A}$; for the implication below, couple the reduction and the game only on the buyer states of valid-witness queries, which are identical.

Suppose $G_{\text{sig},10}$ outputs one. The pair (m^*, σ^*) is a valid signature forgery. Consider any record $(\text{pk}_{\text{LE}}, \tilde{\sigma}_{\text{LE}}) \in \mathcal{T}_{\text{u}}[m^*]$ with $\text{pk}_{\text{LE}} \notin \mathcal{C}_{\text{u}}$. Such a record was created only for a compiler query satisfying $(\text{stmt}, \widehat{\text{wit}}) \in \text{Rel}$. If

$$\text{sk}'_{\text{LE}} = \text{AS.Extract}(\text{vk}, \tilde{\sigma}_{\text{LE}}, \sigma^*, \text{pk}_{\text{LE}})$$

satisfied $(\text{pk}_{\text{LE}}, \text{sk}'_{\text{LE}}) \in \text{Rel}_{\text{LE}}$, correctness of **LE** and the equality guaranteed outside **BadExt** would give

$$\text{AS}^{\text{Ad}}.\text{Extract}(\text{st}_B, \sigma^*) = \text{LE.Dec}(\text{sk}'_{\text{LE}}, c_{\text{LE}}) = \widehat{\text{wit}}.$$

Here we use correctness of **LE** for every secret key related to pk_{LE} by Rel_{LE} , as is the case for the key relation of the stated instantiation. This contradicts E_{sig} . Thus every underlying record not excluded by $\mathcal{C}_{\text{u}}$ extracts an invalid witness, so $\mathcal{B}_{\text{sig}}$ wins the underlying extractability experiment. Accounting for **BadKeys** and **BadExt** gives

$$\Pr[G_{\text{sig},10} = 1] \leq \Pr[\text{Ext}_{\mathcal{B}_{\text{sig}}, \text{AS}}(\lambda) = 1] + \nu(\lambda).$$

Combining the last two bounds proves the lemma. $\square$

5.3.3 Seller Fairness

Proposition 5.7. *Assume **AS** is an adaptable (Definition 3.12) adaptor signature scheme. Suppose that **HE** and **LE** are IND-CPA secure, **HE** has fully re-randomizable ciphertexts and messages, and honest **HE** ciphertexts have negligible maximum point probability. If NIZK_{CE} and NIZK_{HE} are zero-knowledge, then the AS^{Ad} scheme in Figure 8 has seller fairness as in Definition 4.5.*

Proof. To prove this statement, we define the following sequence of games:

- G_0 is $\text{FairReal}_{\mathcal{A}, \text{AS}^{\text{Ad}}}(\lambda, \text{stmt}, \text{wit})$.
- G_1 is the same as G_0 , except that the adaptation oracle does not decrypt c_m . Instead, it considers the ordered list $\mathcal{Q}$ of inputs previously queried to H_{mask} by the adversary and searches for the first $q \in \mathcal{Q}$ such that

$$\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m.$$

If such a query exists, the oracle sets $s := q$ and continues as in G_0 ; otherwise it returns **bad**.

- G_2 is the same as G_1 , except that, after finding s , the oracle computes

$$k_L := H_{\text{key}}(s), \quad \tilde{\sigma}_{\text{LE}} := \text{SKE.Dec}(k_L, \text{ct})$$

and sets $b := 0$ and returns **good** if and only if

$$\text{AS.pVf}(\text{vk}, m, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}) = 1.$$

Otherwise it returns **bad**. In particular, the oracle no longer invokes AS.Adapt and no longer uses either sk_{HE} or sk_{LE} .

- G_3 is the same as G_2 , except that the CRS and proof π_{CE} are generated using the simulator of NIZK_{CE} .

- G_4 is the same as G_3 , except that the CRS and proof π_{HE} are generated using the simulator of NIZK_{HE} .
- G_5 is the same as G_4 , except that AdGen simultaneously replaces

$$(c_{\text{HE}}, c_{\text{LE}}) = (\text{HE.Enc}(\text{pk}_{\text{HE}}, \text{wit}), \text{LE.Enc}(\text{pk}_{\text{LE}}, \text{wit}))$$

with

$$(c_{\text{HE}}, c_{\text{LE}}) = (\text{HE.Enc}(\text{pk}_{\text{HE}}, 0^\ell), \text{LE.Enc}(\text{pk}_{\text{LE}}, 0^\ell)).$$

Notice that in G_5 , the algorithms and oracles are those in Figure 9 and can be simulated entirely without knowledge of the witness. We define these algorithms as the simulator Sim of FairIdeal and conclude that $G_5 \equiv \text{FairIdeal}$. As setup is implicit in the fairness experiments, we fold the simulated setup into Sim_1 : the simulated CRSs become the public parameters and the simulator retains their trapdoors. In the random-oracle instantiation, this is implemented by programming the corresponding domain-separated CRS oracle answers.

$\text{Sim}_1(1^\lambda, \text{stmt})$	$\mathcal{O}_P(\text{st}, \text{vk}, \tilde{\sigma}, m)$
$(\text{crs}_{\text{HE}}, \tau_{\text{HE}}) \leftarrow \text{NIZK}_{\text{HE}}.\text{Sim}_1(1^\lambda)$	if $b = 0$: then return $\perp$
$(\text{crs}_{\text{CE}}, \tau_{\text{CE}}) \leftarrow \text{NIZK}_{\text{CE}}.\text{Sim}_1(1^\lambda)$	$\text{pk}_{\text{LE}}, \text{pk}_{\text{HE}}, c^* = \text{st}$
$\text{pk}_{\text{HE}}, \text{sk}_{\text{HE}} \leftarrow \text{HE.KGen}(1^\lambda; r_k)$	$c_m, \text{ct} = \tilde{\sigma}$
$\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}} \leftarrow \text{LE.KGen}(1^\lambda)$	if $\exists q \in \mathcal{Q} : \text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m$
$c_{\text{HE}} := \text{HE.Enc}(\text{pk}_{\text{HE}}, 0^\ell; r_{\text{HE}})$	then $s :=$ the first such q
$c_{\text{LE}} := \text{LE.Enc}(\text{pk}_{\text{LE}}, 0^\ell; r_{\text{LE}})$	else return bad
$x = (\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}})$	$k_L := H_{\text{key}}(s)$
$\pi_{\text{HE}} \leftarrow \text{NIZK}_{\text{HE}}.\text{Sim}_2(\text{crs}_{\text{HE}}, \tau_{\text{HE}}, \text{pk}_{\text{HE}})$	$\tilde{\sigma}_{\text{LE}} = \text{SKE.Dec}(k_L, \text{ct})$
$\pi_{\text{CE}} \leftarrow \text{NIZK}_{\text{CE}}.\text{Sim}_2(\text{crs}_{\text{CE}}, \tau_{\text{CE}}, x)$	if $\text{AS.pVf}(\text{vk}, m, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}) = 1$
encode $1 - \text{Rel}(\text{stmt}, \cdot)$ as F	$b \leftarrow 0$;
$c^* = \text{HE.Eval}(\text{pk}_{\text{HE}}, F, c_{\text{HE}})$	return good;
$\text{adv}_t = (x, \pi_{\text{CE}}, \pi_{\text{HE}})$	return bad
$\text{st} = (\text{pk}_{\text{LE}}, \text{pk}_{\text{HE}}, c^*)$	
return $(\text{adv}_t, \text{st})$	

Figure 9: The description of Sim .

Now, we prove the indistinguishability of the games, relying on the following properties:

- $G_0 \stackrel{c}{\approx} G_1$ by re-randomizability of HE and the negligible maximum point probability of honest HE ciphertexts;
- $G_1 \equiv G_2$ by adaptability of AS ;
- $G_2 \stackrel{c}{\approx} G_3$ by zero-knowledge of NIZK_{CE} ;
- $G_3 \stackrel{c}{\approx} G_4$ by zero-knowledge of NIZK_{HE} ; and
- $G_4 \stackrel{c}{\approx} G_5$ by IND-CPA security of the product of HE and LE .

The formal claims for these indistinguishabilities are stated and proven below. $\square$

Claim 5.8. $G_0 \stackrel{c}{\approx} G_1$

Proof. Fix an invocation of **Adapt** on input $(\text{vk}, \tilde{\sigma}, m)$ with $\tilde{\sigma} = (c_m, \text{ct})$ and internal state $(\text{sk}_{\text{LE}}, \text{pk}_{\text{LE}}, \text{sk}_{\text{HE}}, \text{pk}_{\text{HE}}, c^*)$. In G_0 , the $\mathcal{O}_P$ oracle computes $s := \text{HE.Dec}(\text{sk}_{\text{HE}}, c_m)$ and then deterministically recomputes $\tilde{c}_m = \text{Mask}(\text{pk}_{\text{HE}}, c^*, s)$, and aborts if $\tilde{c}_m \neq c_m$.

In G_1 , the oracle does not decrypt c_m and instead checks whether there exists $q \in \mathcal{Q}$ (the set of inputs on which the adversary has queried H_{mask}) such that $\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m$. If found, it sets $s := q$ and proceeds identically to G_0 .

The two oracles return different outcomes only if either no queries are valid, i.e., $\forall q \in \mathcal{Q}, \text{Mask}(\text{pk}_{\text{HE}}, c^*, q) \neq c_m$, or there are (at least) two queries $q \neq q'$ such that $\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = \text{Mask}(\text{pk}_{\text{HE}}, c^*, q') = c_m$.

Notice that

$$\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = \text{CtRand}(\text{Enc}(\text{pk}_{\text{HE}}, q; 0) \boxplus \text{MsgRand}(c^*; u_r); u_a),$$

where (u_r, u_a) are derived from $H_{\text{mask}}(q)$, and $\boxplus$ denotes homomorphic addition $\text{HE.HomAdd}(\text{pk}_{\text{HE}}, \cdot, \cdot)$.

Since in the fairness game we have a valid witness wit , then it means that c^* always encrypts 0, so also $\text{HE.MsgRand}(c^*; u_r)$ will be an encryption of 0 by correctness of MsgRand . Thus $c_a = \text{HE.Enc}(\text{pk}_{\text{HE}}, q; 0) \boxplus \text{HE.MsgRand}(c^*; u_r)$ is an encryption of q for which we know the randomness u , i.e. $c_a = \text{Enc}(\text{pk}_{\text{HE}}, q; u)$.

Now, observe that for any q we have $\text{Dec}(\text{sk}_{\text{HE}}, \text{Mask}(\text{pk}_{\text{HE}}, c^*, q)) = q$, so we cannot have the second case where two queries collide producing the same c_m . In particular, the mismatches between G_0 and G_1 can then happen only when:

- $\tilde{c}_m = c_m$, so G_0 does not abort, but there is no query q .
- $\tilde{c}_m \neq c_m$, so G_0 aborts, but there is a query q with $\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m$.

Notice again that in the second case

$$q = \text{Dec}(\text{sk}_{\text{HE}}, \text{Mask}(\text{pk}_{\text{HE}}, c^*, q)) = \text{Dec}(\text{sk}_{\text{HE}}, c_m) = s,$$

so we would have $\tilde{c}_m = \text{Mask}(\text{pk}_{\text{HE}}, c^*, s) = \text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m$, which is a contradiction.

We are now left with the case where the adversary produced a valid $\tilde{c}_m = c_m$, but he never queried the H_{mask} oracle at a point q where $\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m$, meaning that it never queried at $s = \text{Dec}(\text{sk}_{\text{HE}}, c_m)$.

Now, by our earlier analysis, $\tilde{c}_m$ is equal to

$$\text{CtRand}(\text{Enc}(\text{pk}_{\text{HE}}, s; u); u_a).$$

The value u_a is uniform and independent of the adversary's view whenever the adversary has not queried H_{mask} on s . Let $\Delta_{\text{ct}}(\lambda)$ denote the statistical distance in the ciphertext re-randomization property, and let

$$\mu_{\text{HE}}(\lambda) := \max_{\text{pk}_{\text{HE}}, m, c} \Pr[\text{HE.Enc}(\text{pk}_{\text{HE}}, m) = c],$$

where the maximum ranges over honestly generated public keys and valid messages and ciphertexts. By assumption, μ_{HE} is negligible. Full ciphertext re-randomizability gives

$$\text{CtRand}(\text{pk}_{\text{HE}}, \text{Enc}(\text{pk}_{\text{HE}}, s; u); u_a) \stackrel{s}{\approx} \text{Enc}(\text{pk}_{\text{HE}}, s).$$

Hence, conditioned on the previous view and on s not having been queried to H_{mask} , the probability that the fresh value on the left is equal to the adversarially fixed ciphertext c_m is at most

$$\mu_{\text{HE}}(\lambda) + \Delta_{\text{ct}}(\lambda).$$

If the adversary makes at most q_P adaptation queries, a union bound shows that the probability that G_0 and G_1 ever diverge is at most

$$q_P(\mu_{\text{HE}}(\lambda) + \Delta_{\text{ct}}(\lambda)),$$

which is negligible. $\square$

Claim 5.9. $G_1 \equiv G_2$.

Proof. Couple the two games using the same randomness and random-oracle tables. Consider an adaptation query $(\text{vk}, \tilde{\sigma}, m)$, where $\tilde{\sigma} = (c_m, \text{ct})$. Both games first search the same ordered list $\mathcal{Q}$ of adversarial H_{mask} -queries for the first value q satisfying $\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m$. If no such value exists, both games return **bad**. Otherwise, both set $s := q$ and compute the same values

$$k_L := H_{\text{key}}(s) \quad \text{and} \quad \tilde{\sigma}_{\text{LE}} := \text{SKE.Dec}(k_L, \text{ct}).$$

Suppose first that

$$\text{AS.pVf}(\text{vk}, m, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}) = 0.$$

By the definition of $\text{AS}^{\text{Ad}}.\text{Adapt}$, game G_1 returns $\perp$ before invoking the underlying adaptation algorithm. Hence the outer verification fails and $\mathcal{O}_P$ returns **bad**, exactly as in G_2 .

Suppose instead that

$$\text{AS.pVf}(\text{vk}, m, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}) = 1.$$

The pair $(\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}})$ was generated honestly and is therefore a valid statement-witness pair for the relation of the underlying adaptor signature. Pre-signature adaptability of AS gives

$$\Sigma.\text{Vf}(\text{vk}, m, \text{AS.Adapt}(\text{vk}, \tilde{\sigma}_{\text{LE}}, \text{sk}_{\text{LE}})) = 1.$$

This implication applies even if vk and $\tilde{\sigma}_{\text{LE}}$ were chosen maliciously, because Definition 3.12 quantifies over arbitrary verification keys and arbitrary pre-signatures. Consequently, G_1 returns **good** and sets its control flag to zero, exactly as G_2 does.

Thus the response and the control flag agree after every query. Induction over the adaptive queries of $\mathcal{A}$ shows that the complete views in G_1 and G_2 are identically distributed. $\square$

Claim 5.10. $G_2 \stackrel{c}{\approx} G_3$

Proof. Suppose that $\mathcal{D}$ distinguishes the two games with non-negligible advantage. We construct a zero-knowledge adversary $\mathcal{B}_{\text{CE}}$ against NIZK_{CE} . The reduction receives from its challenger a CRS that is either honestly generated or generated by the zero-knowledge simulator. It generates both encryption keypairs and the ciphertexts

$$c_{\text{HE}} \leftarrow \text{HE.Enc}(\text{pk}_{\text{HE}}, \text{wit}; r_{\text{HE}}), \quad c_{\text{LE}} \leftarrow \text{LE.Enc}(\text{pk}_{\text{LE}}, \text{wit}; r_{\text{LE}})$$

honestly. It then submits

$$x = (\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}})$$

and the known witness $(\text{wit}, r_{\text{HE}}, r_{\text{LE}})$ to its challenge proof oracle and uses the returned proof as π_{CE} . The reduction generates π_{HE} honestly, computes c^* , simulates the random oracles by lazy sampling, and answers adaptation queries using the public oracle of G_2 .

The statement x belongs to $\mathcal{L}_{\text{CE}}$, since both ciphertexts encrypt the same bit vector wit . If the zero-knowledge challenger uses the honest CRS and prover, the reduction produces View_2 ; if it uses the simulated CRS and prover, the reduction produces View_3 . The reduction outputs $\mathcal{D}$'s decision. Therefore

$$|\Pr[\mathcal{D}(\text{View}_2) = 1] - \Pr[\mathcal{D}(\text{View}_3) = 1]| \leq \text{Adv}_{\text{NIZK}_{\text{CE}}, \mathcal{B}_{\text{CE}}}^{\text{zk}}(\lambda),$$

which is negligible. For the Fiat–Shamir instantiation, the same reduction uses the simulator from Lemma A.3 and programs the domain-separated Fiat–Shamir oracle in the standard way. $\square$

Claim 5.11. $G_3 \stackrel{\mathcal{C}}{\approx} G_4$

Proof. This is a direct reduction to zero knowledge of NIZK_{HE} . Suppose that $\mathcal{D}$ distinguishes G_3 and G_4 with non-negligible advantage. A zero-knowledge adversary $\mathcal{B}_{\text{HE}}$ obtains from its challenger a CRS for NIZK_{HE} . It samples

$$(\text{pk}_{\text{HE}}, \text{sk}_{\text{HE}}) \leftarrow \text{HE.KGen}(1^\lambda; r_k)$$

and submits the true statement pk_{HE} , together with the witness $(\text{sk}_{\text{HE}}, r_k)$, to the challenge proof oracle. Independently, it generates the simulated CRS and proof for NIZK_{CE} as in G_3 , generates the LE keypair and both encryptions of wit , and completes the advertisement. It then simulates the random oracles and the public adaptation oracle of G_2 .

An honest response from the NIZK_{HE} challenger yields View_3 , whereas a simulated response yields View_4 . Hence

$$|\Pr[\mathcal{D}(\text{View}_3) = 1] - \Pr[\mathcal{D}(\text{View}_4) = 1]| \leq \text{Adv}_{\text{NIZK}_{\text{HE}}, \mathcal{B}_{\text{HE}}}^{\text{zk}}(\lambda),$$

which is negligible. Observe that $\text{pk}_{\text{HE}} \in \mathcal{L}_{\text{HE}}$ because it is honestly generated, so the reduction only requests a proof for a true statement. After this transition, neither the proof generation nor the adaptation oracle requires sk_{HE} ; moreover, the oracle has not required sk_{LE} since G_2 . The two secret keys may therefore be omitted from the internal state in all subsequent games. $\square$

Claim 5.12. $G_4 \stackrel{\mathcal{C}}{\approx} G_5$

Proof. Recall that the product of two IND-CPA-secure encryption schemes is IND-CPA secure. In our setting, define the product encryption scheme $\text{PKE}_\times$ by independently generating an HE keypair and an LE keypair and setting

$$\text{Enc}_\times((\text{pk}_{\text{HE}}, \text{pk}_{\text{LE}}), m) := (\text{HE.Enc}(\text{pk}_{\text{HE}}, m), \text{LE.Enc}(\text{pk}_{\text{LE}}, m)),$$

with independent encryption randomness. Its IND-CPA security follows from a two-step hybrid which first changes the HE component and then the LE component. Since the witness is encrypted componentwise, we use the componentwise extension of this product scheme; its security follows from ordinary IND-CPA by the standard hybrid over the ℓ coordinates.

Suppose now that $\mathcal{D}$ distinguishes G_4 and G_5 with non-negligible advantage. We construct an IND-CPA adversary $\mathcal{B}_\times$ against $\text{PKE}_\times$. The reduction receives the independently generated public keys $(\text{pk}_{\text{HE}}, \text{pk}_{\text{LE}})$ from its challenger and submits the equal-length challenge messages

$$m_0 := \text{wit} \quad \text{and} \quad m_1 := 0^\ell.$$

It receives the challenge ciphertext

$$(c_{\text{HE}}, c_{\text{LE}}) = (\text{HE.Enc}(\text{pk}_{\text{HE}}, m_\beta), \text{LE.Enc}(\text{pk}_{\text{LE}}, m_\beta))$$

for the hidden bit β .

The reduction generates simulated CRSs, retaining the simulation trapdoors τ_{CE} and τ_{HE} , and sets

$$x := (\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}}).$$

It computes

$$\pi_{\text{CE}} \leftarrow \text{NIZK}_{\text{CE}}.\text{Sim}_2(\text{crs}_{\text{CE}}, \tau_{\text{CE}}, x), \quad \pi_{\text{HE}} \leftarrow \text{NIZK}_{\text{HE}}.\text{Sim}_2(\text{crs}_{\text{HE}}, \tau_{\text{HE}}, \text{pk}_{\text{HE}}),$$

and

$$c^* := \text{HE.Eval}(\text{pk}_{\text{HE}}, F, c_{\text{HE}}).$$

It gives $\text{adv} = (x, \pi_{\text{CE}}, \pi_{\text{HE}})$ to $\mathcal{A}$ and simulates H_{mask} and H_{key} by lazy sampling. On an adaptation query $(\text{vk}, (c_m, \text{ct}), m)$, it searches the ordered list of adversarial H_{mask} -queries for the first q such that

$$\text{Mask}(\text{pk}_{\text{HE}}, c^*, q) = c_m.$$

If no such query exists, it returns **bad**. Otherwise it sets $s := q$, computes

$$k_L := H_{\text{key}}(s), \quad \tilde{\sigma}_{\text{LE}} := \text{SKE.Dec}(k_L, \text{ct}),$$

and returns **good** and sets the control flag to zero if and only if

$$\text{AS.pVf}(\text{vk}, m, \tilde{\sigma}_{\text{LE}}, \text{pk}_{\text{LE}}) = 1.$$

Thus $\mathcal{B}_\times$ can answer every oracle query without either challenge secret key. It finally outputs $\mathcal{D}$'s decision on the resulting seller-fairness output.

If $\beta = 0$, both challenge ciphertexts encrypt wit , and the reduction perfectly simulates G_4 . If $\beta = 1$, both encrypt 0^ℓ , and it perfectly simulates G_5 . Moreover, in both cases $x \in \mathcal{L}_{\text{CE}}$: the two ciphertexts encrypt the same bit vector. Also $\text{pk}_{\text{HE}} \in \mathcal{L}_{\text{HE}}$, since the product challenger generates it honestly. Hence both simulated proofs concern true statements; the reduction never needs to simulate a proof for a false ciphertext-equality statement. We obtain

$$|\Pr[\mathcal{D}(\text{View}_4) = 1] - \Pr[\mathcal{D}(\text{View}_5) = 1]| = \text{Adv}_{\text{PKE}_\times, \mathcal{B}_\times}^{\text{ind-cpa}}(\lambda),$$

which is negligible. □

5.3.4 Strong Canonical Signing

Proposition 5.13. *Assume AS is a correct adaptor signature scheme that satisfies strong canonical signing with respect to Σ as in Definition 3.13. Suppose that NIZK_{CE} and NIZK_{HE} are perfectly complete, SKE is perfectly correct for every key in the range of H_{key} , and HE.Eval is deterministic and perfectly correct. Moreover, suppose that the HE operations used by Mask are perfectly plaintext-preserving: for every honestly generated key pair, every honestly evaluated ciphertext c^* encrypting zero, and every s , it holds that*

$$\text{HE.Dec}(\text{sk}_{\text{HE}}, \text{Mask}(\text{pk}_{\text{HE}}, c^*, s)) = s.$$

Then the scheme AS^{Ad} in Figure 8 satisfies strong canonical signing with respect to Σ as in Definition 4.7.

Proof. Fix a security parameter λ , a key pair (sk, vk) in the support of $\Sigma.\text{KGen}(1^\lambda)$, a message m , a valid pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, a pair $(\text{advt}, \text{st}_S)$ in the support of $\text{AdGen}(\text{stmt}, \text{wit})$, and a PPT distinguisher $\mathcal{D}$. Parse

$$\text{advt} = (x, \pi_{\text{CE}}, \pi_{\text{HE}}), \quad x = (\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}})$$

and

$$\text{st}_S = (\text{sk}_{\text{LE}}, \text{pk}_{\text{LE}}, \text{sk}_{\text{HE}}, \text{pk}_{\text{HE}}, c^*).$$

Since $(\text{advt}, \text{st}_S)$ is in the joint support of honest advertisement generation, $(\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}})$ is a valid statement-witness pair for the key relation underlying $\mathcal{L}_{\text{LE}}$.

Consider the canonical execution of AS^{Ad} . Perfect completeness of NIZK_{CE} and NIZK_{HE} ensures that pSign accepts both advertised proofs. Both AdGen and pSign derive $F = 1 - \text{Rel}(\text{stmt}, \cdot)$ from stmt , and deterministic evaluation therefore ensures that pSign recomputes the ciphertext c^* contained in st_S . Since $(\text{stmt}, \text{wit}) \in \text{Rel}$, c^* encrypts zero, and perfect plaintext preservation of Mask implies that c_m decrypts to the value s sampled by pSign . Consequently, Adapt recomputes the same c_m and the same key $k_L = H_{\text{key}}(s)$. Perfect correctness of SKE then recovers the pre-signature $\tilde{\sigma}_{\text{LE}}$ generated by pSign , while correctness of AS ensures that both the pre-verification and final signature-verification checks accept.

It follows that the output of the canonical execution is identically distributed to

$$\tilde{\sigma}_{\text{LE}} \leftarrow \text{AS.pSign}(\text{sk}, m, \text{pk}_{\text{LE}}), \quad \sigma' \leftarrow \text{AS.Adapt}(\text{vk}, \tilde{\sigma}_{\text{LE}}, \text{sk}_{\text{LE}}),$$

which is precisely the canonical-signing distribution of AS for the pair $(\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}})$. By strong canonical signing of AS , this distribution is computationally indistinguishable from $\Sigma.\text{Sign}(\text{sk}, m)$.

Finally, the values $(\text{stmt}, \text{wit}, \text{advt})$ are fixed in both distributions. Any distinguisher for the two distributions in Definition 4.7 therefore yields a distinguisher for strong canonical signing of AS by hard-coding these values and forwarding the signature. This proves the claim. $\square$

5.3.5 Statement Truth Privacy

Proposition 5.14. *Suppose that HE and LE are IND-CPA secure and that NIZK_{CE} and NIZK_{HE} are zero-knowledge. If the statement marginals of D_{LRel}^T and D_{LRel}^F are computationally indistinguishable, then the AS^{Ad} scheme in Figure 8 has statement truth privacy as in Definition 4.6.*

Proof. Define the simulator $\mathcal{S}$ as follows. It generates independently

$$(\text{pk}_{\text{HE}}, \text{sk}_{\text{HE}}) \leftarrow \text{HE.KGen}(1^\lambda; r_k), \quad (\text{pk}_{\text{LE}}, \text{sk}_{\text{LE}}) \leftarrow \text{LE.KGen}(1^\lambda),$$

encrypts the all-zero bit vector under both keys, and sets

$$\begin{aligned} c_{\text{HE}} &\leftarrow \text{HE.Enc}(\text{pk}_{\text{HE}}, 0^\ell; r_{\text{HE}}), & c_{\text{LE}} &\leftarrow \text{LE.Enc}(\text{pk}_{\text{LE}}, 0^\ell; r_{\text{LE}}), \\ x &= (\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}}). \end{aligned}$$

It generates the proofs honestly as

$$\begin{aligned} \pi_{\text{HE}} &\leftarrow \text{NIZK}_{\text{HE}}.\text{Prove}(\text{crs}_{\text{HE}}, \text{pk}_{\text{HE}}, (\text{sk}_{\text{HE}}, r_k)), \\ \pi_{\text{CE}} &\leftarrow \text{NIZK}_{\text{CE}}.\text{Prove}(\text{crs}_{\text{CE}}, x, (0^\ell, r_{\text{HE}}, r_{\text{LE}})), \end{aligned}$$

and outputs

$$\text{adv} = (x, \pi_{\text{CE}}, \pi_{\text{HE}}).$$

The simulator ignores stmt .

We prove indistinguishability using the following hybrids. Hybrid H_0 is the real distribution in Definition 4.6: it samples $(\text{stmt}, \text{wit}) \leftarrow D_{L_{\text{Rel}}}^T$ and generates the advertisement honestly. Hybrid H_1 replaces the CRS and proof π_{CE} by the output of the zero-knowledge simulator for NIZK_{CE} . Since both advertised ciphertexts honestly encrypt the same bit vector wit , the statement x is in $\mathcal{L}_{\text{CE}}$. A distinguisher between H_0 and H_1 therefore directly yields a zero-knowledge adversary against NIZK_{CE} .

Hybrid H_2 additionally replaces the CRS and proof π_{HE} by the output of the zero-knowledge simulator for NIZK_{HE} . The public key pk_{HE} is honestly generated and hence belongs to $\mathcal{L}_{\text{HE}}$. Thus a distinguisher between H_1 and H_2 directly yields a zero-knowledge adversary against NIZK_{HE} .

Hybrid H_3 simultaneously replaces the ciphertext pair

$$(\text{HE.Enc}(\text{pk}_{\text{HE}}, \text{wit}), \text{LE.Enc}(\text{pk}_{\text{LE}}, \text{wit}))$$

by

$$(\text{HE.Enc}(\text{pk}_{\text{HE}}, 0^\ell), \text{LE.Enc}(\text{pk}_{\text{LE}}, 0^\ell)).$$

Recall from the proof of Proposition 5.7 that the product of the componentwise extensions of HE and LE is IND-CPA secure. Suppose that a distinguisher $\mathcal{D}$ separates H_2 and H_3 . An adversary against product IND-CPA first samples $(\text{stmt}, \text{wit}) \leftarrow D_{L_{\text{Rel}}}^T$, receives $(\text{pk}_{\text{HE}}, \text{pk}_{\text{LE}})$ from its challenger, and submits $(\text{wit}, 0^\ell)$ as its challenge messages. Given the challenge pair $(c_{\text{HE}}, c_{\text{LE}})$, it forms $x = (\text{pk}_{\text{HE}}, c_{\text{HE}}, \text{pk}_{\text{LE}}, c_{\text{LE}})$ and simulates both proofs. It then runs $\mathcal{D}$ on the resulting tuple $(1^\lambda, \text{stmt}, \text{adv})$.

If the product challenge encrypts wit , this is exactly H_2 ; if it encrypts 0^ℓ , it is exactly H_3 . In either case, both ciphertexts encrypt the same bit vector, and both public keys are honestly generated. Consequently, $x \in \mathcal{L}_{\text{CE}}$ and $\text{pk}_{\text{HE}} \in \mathcal{L}_{\text{HE}}$ for both challenge bits, so the reduction only simulates proofs for true statements. Distinguishing H_2 from H_3 therefore contradicts product IND-CPA security.

Hybrid H_4 replaces the simulated CRS and proof π_{HE} by an honest CRS and proof for the same public key. Hybrid H_5 likewise replaces the simulated CRS and proof π_{CE} by an honest CRS and proof for the zero ciphertexts. These are the reverse of the zero-knowledge transitions used above. They apply because the game knows $(\text{sk}_{\text{HE}}, r_k)$ and the openings $(0^\ell, r_{\text{HE}}, r_{\text{LE}})$, and both statements are true. Hence $H_3 \approx H_4 \approx H_5$.

Finally, hybrid H_6 replaces the statement sampled from the true-statement marginal by a statement sampled from $D_{L_{\text{Rel}}}^F$, leaving the honest all-zero advertisement unchanged. A distinguisher between H_5 and H_6 immediately distinguishes the statement marginals of $D_{L_{\text{Rel}}}^T$ and $D_{L_{\text{Rel}}}^F$: on receiving its challenge statement, it independently generates the honest all-zero advertisement and invokes $\mathcal{D}$. By assumption this advantage is negligible. Hybrid H_6 is exactly the ideal distribution obtained using $\mathcal{S}$, which concludes the proof. $\square$

6 Adaptor Signatures from Subliminal Channels using Circuit-Private FHE

In this section, we formally show how to realize general adaptor signatures from circuit-private FHE. The informal description of this scheme can be found in Section 2.2. Before formally presenting our

construction, we describe the different relations for the NIZK proof of the advertisement NIZK_{ad} as well as for the presignature evaluation $\text{NIZK}_{\text{eval}}$. We also formally define the circuit that is evaluated by the FHE scheme.

Proof relations. The advertisement relation is

$$\mathcal{L}_{\text{ad}} := \{((\text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}), (\text{sk}_{\text{FHE}}, \rho_K, \text{wit}, \rho_F)) : \\ \text{FHE.KeyGen}(1^\lambda; \rho_K) = (\text{pk}_{\text{FHE}}, \text{sk}_{\text{FHE}}) \wedge \text{ct}_{\text{wit}} = \text{FHE.Enc}(\text{pk}_{\text{FHE}}, \text{wit}; \rho_F)\}.$$

This relation establishes only well-formedness and knowledge of the encrypted plaintext. It deliberately does not assert $(\text{stmt}, \text{wit}) \in \text{Rel}$; validity of the witness is checked inside the evaluated circuit.

The evaluation relation is

$$\mathcal{L}_{\text{ev}} := \{((\text{vk}_\Sigma, m, \text{stmt}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \text{ct}_\sigma), (\text{sk}_\Sigma, k_{\text{SKE}}, \rho_{\text{SKE}}, r_E, r_{\text{ev}})) : \\ \Sigma.\text{DerVerKey}(\text{sk}_\Sigma) = \text{vk}_\Sigma \wedge \text{SKE.KGen}(1^\lambda; \rho_{\text{SKE}}) = k_{\text{SKE}} \wedge \\ \theta = (\text{sk}_\Sigma, k_{\text{SKE}}, m, \text{stmt}, r_E) \wedge \text{ct}_\sigma = \text{FHE.Eval}(\text{pk}_{\text{FHE}}, C_\theta, \text{ct}_{\text{wit}}; r_{\text{ev}})\}.$$

For the FHE scheme, we have the following circuit that is being evaluated: Let sk_Σ be a signing key, k_{SKE} an encryption key, a message m , statement stmt , and encryption coins r_E , set

$$\theta := (\text{sk}_\Sigma, k_{\text{SKE}}, m, \text{stmt}, r_E),$$

then the circuit is defined as

$$C_\theta(\text{wit}') := \begin{cases} \Sigma.\text{Sign}(\text{sk}_\Sigma, m; \text{SKE.Enc}(k_{\text{SKE}}, \text{wit}'; r_E)), & (\text{stmt}, \text{wit}') \in \text{Rel}, \\ \perp, & (\text{stmt}, \text{wit}') \notin \text{Rel}. \end{cases}$$

The formal description of our scheme can be found in Figure 10. It achieves the following properties.

Theorem 6.1. *Let Rel be a hard NP relation. Let Σ be a perfectly correct and strongly unforgeable digital signature scheme that supports verification-key derivation and provides a perfectly correct subliminal channel with respect to SKE. Let SKE be a perfectly correct secret-key encryption scheme with pseudorandom ciphertexts. Suppose that the FHE scheme is perfectly correct, IND-CPA secure, and is statistically circuit private. Suppose NIZK_{ad} is perfectly complete, zero knowledge, and knowledge sound, and that NIZK_{ev} is perfectly complete, zero knowledge, and sound. Then the construction in Figure 10 is correct, pre-signature adaptable, extractable, and strong canonical signing with respect to Σ . If, in addition, the prescribed true- and false-statement distributions are computationally indistinguishable, the construction also satisfies distributional statement truth privacy.*

We prove this theorem by splitting it into multiple lemmas over the next sections.

Setup(1^λ) $\text{crs}_{\text{ad}} \leftarrow \text{NIZK}_{\text{ad}}.\text{Setup}(1^\lambda, \mathcal{L}_{\text{ad}})$ $\text{crs}_{\text{ev}} \leftarrow \text{NIZK}_{\text{ev}}.\text{Setup}(1^\lambda, \mathcal{L}_{\text{ev}})$ return $(\text{crs}_{\text{ad}}, \text{crs}_{\text{ev}})$ AdGen(stmt, wit) $\rho_K \leftarrow \$\{0, 1\}^{\ell_K}$ $(\text{pk}_{\text{FHE}}, \text{sk}_{\text{FHE}})$ $\leftarrow \text{FHE}.\text{KeyGen}(1^\lambda; \rho_K)$ $\rho_F \leftarrow \$\{0, 1\}^{\ell_F}$ $\text{ct}_{\text{wit}} \leftarrow \text{FHE}.\text{Enc}(\text{pk}_{\text{FHE}}, \text{wit}; \rho_F)$ $x_{\text{ad}} \leftarrow (\text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}})$ $w_{\text{ad}} \leftarrow (\text{sk}_{\text{FHE}}, \rho_K, \text{wit}, \rho_F)$ $\pi_{\text{ad}} \leftarrow \text{NIZK}_{\text{ad}}.\text{Prove}(\text{crs}_{\text{ad}}, x_{\text{ad}}, w_{\text{ad}})$ $\text{adv}_t \leftarrow (\text{stmt}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \pi_{\text{ad}})$ $\text{st}_S \leftarrow (\text{stmt}, \text{sk}_{\text{FHE}})$ return $(\text{adv}_t, \text{st}_S)$ Adapt($\text{vk}_\Sigma, m, \tilde{\sigma}, \text{st}_S, \text{wit}$) parse $\tilde{\sigma} = (\text{ct}_\sigma, \pi_{\text{ev}})$ parse $\text{st}_S = (\text{stmt}, \text{sk}_{\text{FHE}})$ if $(\text{stmt}, \text{wit}) \notin \text{Rel}$ then return $\perp$ $\sigma \leftarrow \text{FHE}.\text{Dec}(\text{sk}_{\text{FHE}}, \text{ct}_\sigma)$ if $\Sigma.\text{Verify}(\text{vk}_\Sigma, m, \sigma) = 1$ then return σ return $\perp$ Extract(st_B, σ) parse $\text{st}_B = (\text{sk}_\Sigma, k_{\text{SKE}}, m, \text{stmt})$ $\text{wit}' \leftarrow \Sigma.\text{SubExt}(\text{sk}_\Sigma, m, \sigma, k_{\text{SKE}})$ if $(\text{stmt}, \text{wit}') \in \text{Rel}$ then return wit' return $\perp$	pSign($\text{sk}_\Sigma, m, \text{adv}_t, \text{stmt}$) parse $\text{adv}_t = (\text{stmt}_{\text{ad}}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \pi_{\text{ad}})$ assert $\text{stmt}_{\text{ad}} = \text{stmt}$ $x_{\text{ad}} \leftarrow (\text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}})$ assert $\text{NIZK}_{\text{ad}}.\text{Verify}(\text{crs}_{\text{ad}}, x_{\text{ad}}, \pi_{\text{ad}}) = 1$ $\text{vk}_\Sigma \leftarrow \Sigma.\text{DerVerKey}(\text{sk}_\Sigma)$ assert $\text{vk}_\Sigma \neq \perp$ $\rho_{\text{SKE}} \leftarrow \$\{0, 1\}^{\ell_{\text{SKE}}}$ $k_{\text{SKE}} \leftarrow \text{SKE}.\text{KGen}(1^\lambda; \rho_{\text{SKE}})$ $r_E \leftarrow \$\{0, 1\}^{\ell_E}; \quad r_{\text{ev}} \leftarrow \$\{0, 1\}^{\ell_{\text{ev}}}$ $\theta \leftarrow (\text{sk}_\Sigma, k_{\text{SKE}}, m, \text{stmt}, r_E)$ $\text{ct}_\sigma \leftarrow \text{FHE}.\text{Eval}(\text{pk}_{\text{FHE}}, C_\theta, \text{ct}_{\text{wit}}; r_{\text{ev}})$ $x_{\text{ev}} \leftarrow (\text{vk}_\Sigma, m, \text{stmt}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \text{ct}_\sigma)$ $w_{\text{ev}} \leftarrow (\text{sk}_\Sigma, k_{\text{SKE}}, \rho_{\text{SKE}}, r_E, r_{\text{ev}})$ $\pi_{\text{ev}} \leftarrow \text{NIZK}_{\text{ev}}.\text{Prove}(\text{crs}_{\text{ev}}, x_{\text{ev}}, w_{\text{ev}})$ $\tilde{\sigma} \leftarrow (\text{ct}_\sigma, \pi_{\text{ev}})$ $\text{st}_B \leftarrow (\text{sk}_\Sigma, k_{\text{SKE}}, m, \text{stmt})$ return $(\tilde{\sigma}, \text{st}_B)$ pVf($\text{vk}_\Sigma, m, \tilde{\sigma}, \text{adv}_t, \text{stmt}$) parse $\tilde{\sigma} = (\text{ct}_\sigma, \pi_{\text{ev}})$ parse $\text{adv}_t = (\text{stmt}_{\text{ad}}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \pi_{\text{ad}})$ if $\text{stmt}_{\text{ad}} \neq \text{stmt}$ then return 0 $x_{\text{ad}} \leftarrow (\text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}})$ if $\text{NIZK}_{\text{ad}}.\text{Verify}(\text{crs}_{\text{ad}}, x_{\text{ad}}, \pi_{\text{ad}}) = 0$, then return 0 $x_{\text{ev}} \leftarrow (\text{vk}_\Sigma, m, \text{stmt}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \text{ct}_\sigma)$ return $\text{NIZK}_{\text{ev}}.\text{Verify}(\text{crs}_{\text{ev}}, x_{\text{ev}}, \pi_{\text{ev}})$
--	--

Figure 10: FHE adaptor signatures from the subliminal-channels.

6.1 Correctness

Lemma 6.2 (Correctness). *If the FHE, both NIZK systems, Σ , SKE, and the subliminal decoder are perfectly correct, and Σ supports verification-key derivation according to Definition 3.3, then the construction satisfies correctness as well (Definition 4.2).*

Proof. Let $(\text{stmt}, \text{wit}) \in \text{Rel}$ and consider an honest execution, then the perfect completeness of NIZK leads to the acceptance of pVf . From the perfect FHE correctness, it follows that adaptation decrypts to

$$\sigma = \Sigma.\text{Sign}(\text{sk}_\Sigma, m; \text{SKE}.\text{Enc}(k_{\text{SKE}}, \text{wit}; r_E)).$$

The signature correctness then leads to the verification of σ . Finally, the perfect correctness of the subliminal-channel Definition 3.6 gives

$$\Sigma.\text{SubExt}(\text{sk}_\Sigma, m, \sigma, k_{\text{SKE}}) = \text{wit},$$

which results in the Extract procedure outputting a valid witness for stmt . $\square$

6.2 Pre-Signature Adaptability

Lemma 6.3 (Pre-signature adaptability). *If NIZK_{ad} is perfectly complete, NIZK_{ev} is adaptively sound, the FHE is perfectly correct, and Σ supports verification-key derivation according to Definition 3.3, then the construction satisfies pre-signature adaptability according to Definition 4.4.*

Proof. Fix a satisfying pair $(\text{stmt}, \text{wit}) \in \text{Rel}$ and let $\mathcal{A}$ be a PPT adaptability adversary. We define the adaptability-failure event that we need to bound as

$$\begin{aligned} E_{\text{adapt}} &:= \left\{ \omega \mid \begin{array}{l} \text{pVf}(\text{vk}_\Sigma, m, \tilde{\sigma}, \text{advt}, \text{stmt}) = 1, \\ \Sigma.\text{Verify}(\text{vk}_\Sigma, m, \sigma) = 0 \end{array} \right\} \\ &= \left\{ \omega \mid \begin{array}{l} \text{pVf}(\text{vk}_\Sigma, m, (\text{ct}_\sigma, \pi_{\text{ev}}), (\text{stmt}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \pi_{\text{ad}})) = 1, \\ \Sigma.\text{Verify}(\text{vk}_\Sigma, m, \text{FHE}.\text{Dec}(\text{sk}_{\text{FHE}}, \text{ct}_\sigma)) = 0 \end{array} \right\} \\ &= \left\{ \omega \mid \begin{array}{l} x_{\text{ad}} := (\text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}) \\ x_{\text{ev}} := (\text{vk}_\Sigma, m, \text{stmt}, \text{pk}_{\text{FHE}}, \text{ct}_{\text{wit}}, \text{ct}_\sigma) \\ \text{NIZK}_{\text{ad}}.\text{Verify}(\text{crs}_{\text{ad}}, x_{\text{ad}}, \pi_{\text{ad}}) = 1, \\ \text{NIZK}_{\text{ev}}.\text{Verify}(\text{crs}_{\text{ev}}, x_{\text{ev}}, \pi_{\text{ev}}) = 1, \\ \Sigma.\text{Verify}(\text{vk}_\Sigma, m, \text{FHE}.\text{Dec}(\text{sk}_{\text{FHE}}, \text{ct}_\sigma)) = 0 \end{array} \right\} \end{aligned}$$

The probability of this event is taken over the following experiment:

$$\begin{aligned} (\text{crs}_{\text{ad}}, \text{crs}_{\text{ev}}) &\leftarrow \text{Setup}(1^\lambda), \\ (\text{advt}, \text{st}_S) &\leftarrow \text{AdGen}(\text{stmt}, \text{wit}), \\ (\text{vk}_\Sigma, m, \tilde{\sigma}) &\leftarrow \mathcal{A}(\text{advt}, \text{stmt}), \\ \sigma &\leftarrow \text{Adapt}(\text{vk}_\Sigma, m, \tilde{\sigma}, \text{st}_S, \text{wit}). \end{aligned}$$

Thus, the first two conditions state that the advertisement proof and the evaluation proof are accepted, respectively, capturing the acceptance of the pre-signature. The last condition states that decrypting ct_σ yields a signature that does not verify and, therefore, capturing a failure of adaptation. Since the advertisement is honestly generated, it holds that $x_{\text{ad}} \in \mathcal{L}_{\text{ad}}$, and perfect

completeness implies $\text{NIZK}_{\text{ad}}.\text{Verify}(\text{crs}_{\text{ad}}, x_{\text{ad}}, \pi_{\text{ad}}) = 1$. Therefore, it only remains to distinguish whether the accepted evaluation statement x_{ev} is false or true. Accordingly, we partition $\mathbf{E}_{\text{adapt}}$ into the two events

$$\begin{aligned}\mathbf{E}_{\text{snd}} &:= \{\omega \in \mathbf{E}_{\text{adapt}} \mid x_{\text{ev}} \notin \mathcal{L}_{\text{ev}}\}, \\ \mathbf{E}_{\text{corr}} &:= \{\omega \in \mathbf{E}_{\text{adapt}} \mid x_{\text{ev}} \in \mathcal{L}_{\text{ev}}\}.\end{aligned}$$

The two events are disjoint and exhaustive. Therefore,

$$\Pr[\mathbf{E}_{\text{adapt}}] = \Pr[\mathbf{E}_{\text{snd}}] + \Pr[\mathbf{E}_{\text{corr}}].$$

We show in Lemmas 6.4 and 6.5 that the two summands are negligible, which concludes the proof of the lemma. $\square$

Lemma 6.4 (False evaluation-statement event). *If NIZK_{ev} is sound, then there is a PPT soundness adversary $\mathcal{B}_{\text{snd}}$ such that*

$$\Pr[\mathbf{E}_{\text{snd}}] \leq \nu(\lambda).$$

Proof. A reduction given the NIZK_{ev} CRS generates the remaining public parameters and the honest advertisement, runs $\mathcal{A}$, forms x_{ev} as above, and outputs $(x_{\text{ev}}, \pi_{\text{ev}})$. Membership in $\mathbf{E}_{\text{adapt}}$ implies that pVf accepts and hence that

$$\text{NIZK}_{\text{ev}}.\text{Verify}(\text{crs}_{\text{ev}}, x_{\text{ev}}, \pi_{\text{ev}}) = 1.$$

Whenever $\mathbf{E}_{\text{snd}}$ occurs, we additionally have $x_{\text{ev}} \notin \mathcal{L}_{\text{ev}}$. Thus the reduction outputs an accepting proof for a false statement, which contradicts soundness of the NIZK_{ev} and, therefore, proves the lemma. $\square$

Lemma 6.5 (Valid evaluation-statement event). *If the FHE is perfectly correct and Σ supports verification-key derivation according to Definition 3.3, then*

$$\Pr[\mathbf{E}_{\text{corr}}] = 0.$$

Proof. Suppose that $\mathbf{E}_{\text{corr}}$ occurs. Since $x_{\text{ev}} \in \mathcal{L}_{\text{ev}}$, there exists a language witness

$$\text{wit}'_{\text{ev}} := (\text{sk}'_{\Sigma}, k'_{\text{SKE}}, \rho'_{\text{SKE}}, r'_E, r'_{\text{ev}})$$

such that $\Sigma.\text{DerVerKey}(\text{sk}'_{\Sigma}) = \text{vk}_{\Sigma}$, $\text{SKE.KGen}(1^{\lambda}; \rho'_{\text{SKE}}) = k'_{\text{SKE}}$, and $\text{ct}_{\sigma} = \text{FHE.Eval}(\text{pk}_{\text{FHE}}, C_{\theta'}, \text{ct}_{\text{wit}}; r'_{\text{ev}})$, with $\theta' = (\text{sk}'_{\Sigma}, k'_{\text{SKE}}, m, \text{stmt}, r'_E)$.

We fix one such witness and define

$$\sigma_{\text{out}} := \text{FHE.Dec}(\text{sk}_{\text{FHE}}, \text{ct}_{\sigma}).$$

We split $\mathbf{E}_{\text{corr}}$ into the following two events:

$$\begin{aligned}\mathbf{E}_{\text{FHE}} &:= \{\omega \in \mathbf{E}_{\text{corr}} \mid \sigma_{\text{out}} \neq C_{\theta'}(\text{wit})\}, \\ \mathbf{E}_{\text{vrfy}} &:= \left\{ \omega \in \mathbf{E}_{\text{corr}} \left| \begin{array}{l} \sigma_{\text{out}} = C_{\theta'}(\text{wit}) \wedge \\ \Sigma.\text{Verify}(\text{vk}_{\Sigma}, m, \sigma_{\text{out}}) = 0 \end{array} \right. \right\}.\end{aligned}$$

The condition that $\Sigma.\text{Verify}(\text{vk}_{\Sigma}, m, \sigma_{\text{out}}) = 0$ already follows from the definition of $\mathbf{E}_{\text{adapt}}$. Therefore,

$$\Pr[\mathbf{E}_{\text{corr}}] = \Pr[\mathbf{E}_{\text{FHE}}] + \Pr[\mathbf{E}_{\text{vrfy}}].$$

Event E_{FHE} . Since the advertisement is honestly generated, this implies that pk_{FHE} and ct_{wit} are in the support of the FHE key-generation and the encryption algorithm, respectively, and ct_{wit} encrypts wit under the secret key contained in st_S . From $x_{\text{ev}} \in \mathcal{L}_{\text{ev}}$ it follows that

$$\text{ct}_\sigma = \text{FHE.Eval}(\text{pk}_{\text{FHE}}, C_{\theta'}, \text{ct}_{\text{wit}}; r'_{\text{ev}}).$$

Perfect correctness of FHE implies that

$$\sigma_{\text{out}} = C_{\theta'}(\text{wit}),$$

which, in turn, gives

$$\Pr[E_{\text{FHE}}] = 0.$$

Event E_{vrfy} . In this event, correct decryption of the evaluated ciphertext has already been enforced. Since $(\text{stmt}, \text{wit}) \in \text{Rel}$, the circuit $C_{\theta'}$ generates:

$$\sigma_{\text{out}} = \Sigma.\text{Sign}(\text{sk}'_{\Sigma}, m; \text{SKE.Enc}(k'_{\text{SKE}}, \text{wit}; r'_E)).$$

Given that, $(x_{\text{ev}}, \text{wit}'_{\text{ev}}) \in \text{Rel}_{\text{ev}}$ it is asserted that $\text{vk}_{\Sigma} = \text{DerVerKey}(\text{sk}'_{\Sigma})$. Taking into account Condition (ii) of Definition 3.3 this implies that

$$\Sigma.\text{Verify}(\text{vk}_{\Sigma}, m, \sigma_{\text{out}}) = 1,$$

which, in turn, contradicts the definition of E_{vrfy} , hence

$$\Pr[E_{\text{vrfy}}] = 0.$$

Combining the two bounds yields $\Pr[E_{\text{corr}}] = 0$, which concludes the proof. $\square$

6.3 Extractability

Lemma 6.6 (Extractability). *Suppose Rel is hard, the FHE is perfectly correct, public-key IND-CPA secure and statistically circuit private according to Definition 3.20, NIZK_{ad} satisfies knowledge soundness and zero-knowledge, NIZK_{ev} satisfies zero-knowledge, SKE has pseudorandom ciphertexts, and Σ is strongly unforgeable. Then the construction is extractable (Definition 4.3).*

Proof. Let $\mathcal{A}$ be an extractability adversary and define

$$E_{\text{forge}} := \left\{ \omega \left| \begin{array}{l} \Sigma.\text{Verify}(\text{vk}_{\Sigma}, m^*, \sigma^*) = 1, \\ m^* \notin \mathcal{S}, \text{ and} \\ \forall (\text{adv}, \text{stmt}, \tilde{\sigma}, \text{st}_B) \in \mathcal{T}[m^*] : \\ \text{stmt} \notin \mathcal{X} \implies (\text{stmt}, \text{Extract}(\text{st}_B, \sigma^*)) \notin \text{Rel} \end{array} \right. \right\},$$

as the forgery event that we need to bound.

Then we can partition this into the two events:

$$E_{\text{rel}} := \left\{ \omega \in E_{\text{forge}} \left| \begin{array}{l} \exists (\text{adv}, \text{stmt}, \tilde{\sigma}, \text{st}_B) \in \mathcal{T}[m^*] : \\ \text{stmt} \in \mathcal{X} \wedge (\text{stmt}, \text{Extract}(\text{st}_B, \sigma^*)) \in \text{Rel} \end{array} \right. \right\},$$

$$E_{\text{sig}} := \left\{ \omega \in E_{\text{forge}} \left| \begin{array}{l} \forall (\text{adv}, \text{stmt}, \tilde{\sigma}, \text{st}_B) \in \mathcal{T}[m^*] : \\ (\text{stmt}, \text{Extract}(\text{st}_B, \sigma^*)) \notin \text{Rel} \end{array} \right. \right\}.$$

The condition in E_{forge} rules out successful extraction for statements outside $\mathcal{X}$, so these events are disjoint and exhaustive. Therefore,

$$\Pr[E_{\text{forge}}] = \Pr[E_{\text{rel}}] + \Pr[E_{\text{sig}}].$$

We show in Lemmas 6.7 and 6.8 that the two summands are negligible. $\square$

Lemma 6.7 (Relation-witness event). *Suppose Rel is hard, NIZK_{ad} is zero-knowledge, and the FHE scheme is IND-CPA secure. Let q_Y bound the number of NewStmt queries. Then there is a PPT relation-hardness adversary $\mathcal{B}_{\text{rel}}$ such that*

$$\Pr[E_{\text{rel}}] \leq q_Y \Pr[(\text{stmt}^\dagger, \mathcal{B}_{\text{rel}}(\text{stmt}^\dagger)) \in \text{Rel}] + \nu(\lambda) \leq \nu(\lambda).$$

Proof. The general idea of this proof is to remove the witness from one selected advertisement and then turn this into a reduction against the relation hardness of the chosen instance. Before applying relation hardness, we proceed through the following hybrids.

Hybrid $H_{\text{rel},0}$. This is the real extractability experiment and it outputs 1 exactly on E_{rel} .

Hybrid $H_{\text{rel},1}$. Choose $J \leftarrow [q_Y]$ at the beginning. Output one only if the record witnessing E_{rel} uses the statement returned by the J -th NewStmt query. In Lemma 6.9, we show the following bound:

$$\Pr[H_{\text{rel},0} = 1] \leq q_Y \Pr[H_{\text{rel},1} = 1].$$

Hybrid $H_{\text{rel},2}$. Generate the advertisement CRS with the simulation setup and simulate every honestly generated advertisement proof. The transition between this as the previous hybrid follows from the zero-knowledge property of NIZK_{ad} , as formalized in Lemma 6.10;

$$|\Pr[H_{\text{rel},1} = 1] - \Pr[H_{\text{rel},2} = 1]| \leq \nu(\lambda).$$

Hybrid $H_{\text{rel},3}$. In the selected advertisement, replace $\text{FHE.Enc}(\text{pk}_{\text{FHE}}, \text{wit})$ by $\text{FHE.Enc}(\text{pk}_{\text{FHE}}, 0^{\ell_w})$. Every pSign response is still generated by evaluating the prescribed circuit on the ciphertext appearing in the advertisement and is therefore an efficient transformation of the challenge ciphertext. The transition between this as the previous hybrid follows from the IND-CPA security of the FHE scheme, as formalized in Lemma 6.11:

$$|\Pr[H_{\text{rel},2} = 1] - \Pr[H_{\text{rel},3} = 1]| \leq \nu(\lambda).$$

In the last step, a win in $H_{\text{rel},3}$ directly provides a witness for the selected hard statement: the selected advertisement contains only an encryption of zero, so its simulation does not use that witness. The formal reduction and final bound are given in Lemma 6.12. Combining all bounds gives

$$\begin{aligned} \Pr[E_{\text{rel}}] &= \Pr[H_{\text{rel},0} = 1] \\ &\leq q_Y \Pr[H_{\text{rel},1} = 1] \\ &\leq q_Y \Pr[H_{\text{rel},3} = 1] + \nu(\lambda) \\ &\leq q_Y \Pr[(\text{stmt}^\dagger, \mathcal{B}_{\text{rel}}(\text{stmt}^\dagger)) \in \text{Rel}] + \nu(\lambda). \end{aligned}$$

Here the polynomial factor multiplying the transition errors is absorbed into ν . Relation hardness makes the final expression negligible, which concludes the proof of the lemma. $\square$

Lemma 6.8 (Signature-forgery event). *Suppose NIZK_{ad} is knowledge sound, NIZK_{ev} is zero knowledge, the FHE scheme is perfectly correct and statistically circuit private according to Definition 3.20, the subliminal channel is perfectly correct, SKE has pseudorandom ciphertexts, and Σ is strongly unforgeable. Then there is a PPT strong-unforgeability adversary $\mathcal{B}_{\text{sig}}$ such that*

$$\Pr[\mathbf{E}_{\text{sig}}] \leq \Pr[\text{SigForge}_{\mathcal{B}_{\text{sig}}, \Sigma}(\lambda) = 1] + \nu(\lambda) \leq \nu(\lambda).$$

Proof. The general idea of this proof is to rely on the unforgeability of the signature scheme. To accomplish this, we first have to proceed in a sequence of hybrids replacing the encrypted message to allow for programming using a signature provided by the underlying challenger of the unforgeability game. The hybrids are as follows:

Hybrid $\text{H}_{\text{sig},0}$. This is the real extractability experiment and it outputs 1 exactly on $\mathbf{E}_{\text{sig}}$.

Hybrid $\text{H}_{\text{sig},1}$. Generate the NIZK_{ad} CRS with its extraction setup. For each accepted advertisement, retain the honest witness or extract

$$(\text{sk}_{\text{FHE}}, \rho_K, \widetilde{\text{wit}}, \rho_F)$$

from its proof. Extraction failure and setup indistinguishability change the probability by at most $\nu(\lambda)$, as proved in Lemma 6.13.

For every pSign query whose extracted plaintext is valid, let z_i denote the plaintext output of the prescribed circuit, and let $\mathcal{Z}[m]$ be the multiset of these values for message m . Define

$$\mathbf{E}_{\text{fresh}} := \left\{ \omega \left| \begin{array}{l} \Sigma.\text{Verify}(\text{vk}_{\Sigma}, m^*, \sigma^*) = 1, \\ m^* \notin \mathcal{S}, \quad \text{and} \\ \forall z \in \mathcal{Z}[m^*] : \sigma^* \neq z \end{array} \right. \right\}.$$

The purpose of defining $\mathbf{E}_{\text{fresh}}$ is to capture the exact condition needed by the strong-unforgeability reduction in the final hybrid. Unlike $\mathbf{E}_{\text{sig}}$, it also no longer refers to the extraction states or to subliminal decoding, so the same event can be carried through the remaining hybrids.

From the perfect correctness of the subliminal channel it follows that

$$\mathbf{E}_{\text{sig}} \subseteq \mathbf{E}_{\text{fresh}}.$$

Indeed, if $\sigma^* = z_i$ for a valid query, decoding it with that query's retained key returns $\widetilde{\text{wit}}_i$, contradicting the definition of $\mathbf{E}_{\text{sig}}$. Therefore, it suffices to bound $\mathbf{E}_{\text{fresh}}$, so we focus on this in the remaining hybrids.

We prove the following bound in Lemma 6.13:

$$\Pr[\text{H}_{\text{sig},0} = 1] \leq \Pr[\text{H}_{\text{sig},1} = 1] + \nu(\lambda).$$

Hybrid $\text{H}_{\text{sig},2}$. Replace the NIZK_{ev} CRS by its simulation setup and simulate all evaluation proofs. The transition between this as the previous hybrid follows from the zero-knowledge property of NIZK_{ev} , as formalized in Lemma 6.14,

$$|\Pr[\text{H}_{\text{sig},1} = 1] - \Pr[\text{H}_{\text{sig},2} = 1]| \leq \nu(\lambda).$$

Hybrid $H_{\text{sig},3}$. In this hybrid, we change the pSign answers one at a time. For a query with extracted plaintext $\widetilde{\text{wit}}$, let

$$z_i := \Sigma.\text{Sign}(\text{sk}_\Sigma, m_i; \text{SKE}.\text{Enc}(k_i, \widetilde{\text{wit}}; r_{E,i}))$$

if $(\text{stmt}_i, \widetilde{\text{wit}}) \in \text{Rel}$, and let $z_i := \perp$ otherwise. Replace the evaluated ciphertext by $\text{FHE}.\text{Enc}(\text{pk}_{\text{FHE}}, z_i)$ and retain the simulated proof. The extracted advertisement witness guarantees that the FHE key and input ciphertext are honestly supported. Additionally, we rely on the perfect correctness of the FHE scheme to ensure that the evaluation produces the intended output even if the key pair, as part of the advertisement, and ciphertext, was generated by the adversary $\mathcal{A}$. Thus, z_i is also the plaintext of the evaluated ciphertext in this case.

The transition between this as the previous hybrid follows from the perfect correctness and statistical circuit privacy (Definition 3.20) of the FHE scheme, as formalized in Lemma 6.15,

$$|\Pr[H_{\text{sig},2} = 1] - \Pr[H_{\text{sig},3} = 1]| \leq \nu(\lambda).$$

A response-by-response hybrid covers repeated evaluations of the same advertisement and adaptively chosen queries.

Hybrid $H_{\text{sig},4}$. For every valid query, replace $\text{SKE}.\text{Enc}(k_i, \widetilde{\text{wit}}; r_{E,i})$ by an independent $r_i \leftarrow \$ \{0, 1\}^{\ell_r}$, so that

$$z_i := \Sigma.\text{Sign}(\text{sk}_\Sigma, m_i; r_i).$$

The fresh encryption key used for each query is hidden in st_B , and the event now refers only to z_i , not to the subliminal decoding. The transition between this as the previous hybrid follows from the ciphertext pseudorandomness, as formalized in Lemma 6.16,

$$|\Pr[H_{\text{sig},3} = 1] - \Pr[H_{\text{sig},4} = 1]| \leq \nu(\lambda).$$

In the last step, each valid pSign response in $H_{\text{sig},4}$ is derived from an ordinary signing-oracle response, whereas E_{fresh} explicitly excludes all those responses. Thus a win yields a strong forgery. The formal reduction and final bound are given in Lemma 6.17. Combining all bounds gives

$$\begin{aligned} \Pr[E_{\text{sig}}] &= \Pr[H_{\text{sig},0} = 1] \\ &\leq \Pr[H_{\text{sig},1} = 1] + \nu(\lambda) \\ &\leq \Pr[H_{\text{sig},4} = 1] + \nu(\lambda) \\ &\leq \Pr[\text{SigForge}_{\mathcal{B}_{\text{sig}}, \Sigma}(\lambda) = 1] + \nu(\lambda), \end{aligned}$$

which is negligible by strong unforgeability and concludes the proof of the lemma. $\square$

6.3.1 Proofs of the Hybrid Transitions and Final Bounds.

In this section, we provide the details on the hybrid transitions for the relation-witness event E_{rel} as well as the signature-forgery event E_{sig} .

6.3.2 Bounding the relation-witness event E_{rel} .

Lemma 6.9 (Bounding $H_{\text{rel},1}$). *Let q_Y bound the number of NewStmt queries. Then*

$$\Pr[H_{\text{rel},0} = 1] \leq q_Y \Pr[H_{\text{rel},1} = 1].$$

Proof. We prove the lemma using a standard guessing argument.

For every outcome ω for which $H_{\text{rel},0}$ outputs one, let $I(\omega) \subseteq [q_Y]$ be the nonempty set of indices of NewStmt queries whose returned statement occurs in a record witnessing E_{rel} . Conditioned on this outcome, the selected index belongs to $I(\omega)$ with probability

$$\Pr[J \in I(\omega) \mid \omega] = \frac{|I(\omega)|}{q_Y} \geq \frac{1}{q_Y}.$$

In this case, $H_{\text{rel},1}$ outputs one. Averaging over all outcomes ω and rearranging gives

$$\Pr[H_{\text{rel},0} = 1] \leq q_Y \Pr[H_{\text{rel},1} = 1],$$

which concludes the proof. $\square$

Lemma 6.10 (Indistinguishability of $H_{\text{rel},1}$ and $H_{\text{rel},2}$). *If NIZK_{ad} is zero knowledge, then*

$$|\Pr[H_{\text{rel},1} = 1] - \Pr[H_{\text{rel},2} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by a reduction to the zero-knowledge property of NIZK_{ad} . More precisely, suppose that the probabilities of the two hybrids differ by a non-negligible amount. We construct a zero-knowledge adversary $\mathcal{B}_{\text{zk}}$ that distinguishes the real and simulated experiments of NIZK_{ad} with the same advantage.

The reduction $\mathcal{B}_{\text{zk}}$ receives the NIZK_{ad} CRS from its challenger and runs $\mathcal{A}$. It generates the FHE parameters, the $\text{NIZK}_{\text{eval}}$ CRS, the signature key pair, and every other component of the extractability experiment itself. It also answers the Sign and pSign oracles exactly as in $H_{\text{rel},1}$. Whenever the i -th NewStmt query produces

$$x_{\text{ad},i} = (\text{pk}_{\text{FHE},i}, \text{ct}_{\text{wit},i}) \quad \text{and} \quad w_{\text{ad},i} = (\text{sk}_{\text{FHE},i}, \rho_{K,i}, \text{wit}_i, \rho_{F,i}),$$

$\mathcal{B}_{\text{zk}}$ submits $(x_{\text{ad},i}, w_{\text{ad},i})$ to its adaptive proving oracle and uses the returned proof in the advertisement. Proofs contained in advertisements generated by the adversary are verified under the received NIZK_{ad} CRS. Eventually, $\mathcal{A}$ outputs its forgery attempt, and $\mathcal{B}_{\text{zk}}$ outputs the bit prescribed by the hybrid.

If the underlying challenger uses the real setup and proving algorithm, then $\mathcal{B}_{\text{zk}}$ perfectly simulates $H_{\text{rel},1}$. If the challenger uses the simulation setup and simulates the proofs, then it perfectly simulates $H_{\text{rel},2}$. The $\text{NIZK}_{\text{eval}}$ CRS and all evaluation proofs are honest in both hybrids. Hence a non-negligible difference between the two hybrid probabilities gives a non-negligible advantage against zero knowledge, which proves the claimed bound. $\square$

Lemma 6.11 (Indistinguishability of $H_{\text{rel},2}$ and $H_{\text{rel},3}$). *If the FHE scheme is public-key IND-CPA secure, then*

$$|\Pr[H_{\text{rel},2} = 1] - \Pr[H_{\text{rel},3} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by a reduction to the IND-CPA security of the FHE scheme. In more detail, suppose that $\mathcal{A}$ causes a non-negligible difference between the output probabilities of the two hybrids. We construct an IND-CPA adversary $\mathcal{B}_{\text{FHE}}$ with the same distinguishing advantage.

The reduction receives an FHE public key from its challenger, generates the NIZK_{ad} CRS using the simulation setup, and generates all remaining parameters honestly. When the selected **NewStmt** query is made, it samples $(\text{stmt}_J, \text{wit}_J) \leftarrow \text{RGen}_{\text{Rel}}(1^\lambda)$ and submits $(\text{wit}_J, 0^{\ell_w})$ as its IND-CPA challenge messages. It uses the challenge public key and ciphertext as $(\text{pk}_{\text{FHE},J}, \text{ct}_{\text{wit},J})$ and simulates the advertisement proof using the simulation trapdoor already present in $\text{H}_{\text{rel},2}$.

$\mathcal{B}_{\text{FHE}}$ answers every other oracle query as prescribed by the hybrid. In particular, it can answer every **pSign** query involving the selected advertisement by publicly evaluating the prescribed circuit on the challenge ciphertext. It knows the signature key, the secret-encryption key and coins, and the FHE evaluation coins, and can therefore also generate the corresponding evaluation proofs honestly. It does not need the FHE secret key or an opening of the challenge ciphertext, since the seller state is not given to $\mathcal{A}$. Eventually, it outputs the bit produced by the hybrid.

If the challenge encrypts wit_J , the simulation is $\text{H}_{\text{rel},2}$; if it encrypts 0^{ℓ_w} , it is $\text{H}_{\text{rel},3}$. Thus the difference between the two hybrid probabilities is exactly the distinguishing advantage of $\mathcal{B}_{\text{FHE}}$, which is negligible by the IND-CPA security of FHE. $\square$

Lemma 6.12 (Bounding $\text{H}_{\text{rel},3}$). *There is a PPT relation-hardness adversary $\mathcal{B}_{\text{rel}}$ such that*

$$\Pr[\text{H}_{\text{rel},3} = 1] \leq \Pr[(\text{stmt}^\dagger, \mathcal{B}_{\text{rel}}(\text{stmt}^\dagger)) \in \text{Rel}].$$

Proof. We prove the lemma by a direct reduction to the hardness of Rel . The reduction $\mathcal{B}_{\text{rel}}$ receives a challenge statement $\text{stmt}^\dagger$ sampled from the statement marginal of $\text{RGen}_{\text{Rel}}(1^\lambda)$ and uses it for the selected **NewStmt** query. It generates the NIZK_{ad} CRS with the simulation setup and the evaluation CRS honestly. For the selected advertisement, it generates an FHE key pair, encrypts 0^{ℓ_w} , and simulates the advertisement proof. It simulates the proofs for all other honestly generated advertisements as well and answers every remaining oracle exactly as in $\text{H}_{\text{rel},3}$. In particular, it generates the ordinary-signature key pair and answers each **pSign** query using that fixed signing key and a fresh secret-encryption key.

This simulation does not require a witness for $\text{stmt}^\dagger$: the advertisement relation $\mathcal{L}_{\text{ad}}$ asserts only that the FHE key and ciphertext are honestly supported, not that the encrypted value is a witness for $\text{stmt}^\dagger$. At the end, the reduction scans the records in $\mathcal{T}[m^*]$ whose statement is $\text{stmt}^\dagger$, computes

$$\text{wit}' = \text{Extract}(\text{st}_B, \sigma^*),$$

and returns the first value satisfying $(\text{stmt}^\dagger, \text{wit}') \in \text{Rel}$. Whenever $\text{H}_{\text{rel},3}$ outputs one, its selected record guarantees that such a value exists. Therefore,

$$\Pr[\text{H}_{\text{rel},3} = 1] \leq \Pr[(\text{stmt}^\dagger, \mathcal{B}_{\text{rel}}(\text{stmt}^\dagger)) \in \text{Rel}],$$

which concludes the proof. $\square$

6.3.3 Bounding the signature-forgery event E_{sig} .

Lemma 6.13 (Bounding $\text{H}_{\text{sig},1}$). *If NIZK_{ad} has knowledge soundness and the subliminal channel is perfectly correct, then*

$$\Pr[\text{H}_{\text{sig},0} = 1] \leq \Pr[\text{H}_{\text{sig},1} = 1] + \nu(\lambda).$$

Proof. We prove the lemma in two steps. First, replace the real advertisement CRS with the extraction setup of NIZK_{ad} . If this replacement changed the output probability by a non-negligible amount, the resulting distinguisher would contradict CRS indistinguishability. Thus this step changes the probability by at most $\nu(\lambda)$.

In the resulting experiment, apply the extractor to every accepted adversarial advertisement proof to obtain

$$(\text{sk}_{\text{FHE},i}, \rho_{K,i}, \widetilde{\text{wit}}_i, \rho_{F,i}).$$

For an honestly generated advertisement, retain its known witness. Let Bad_{ext} be the event that an accepted adversarial proof does not yield a full witness in $\mathcal{L}_{\text{ad}}$. If this event occurred with non-negligible probability, the extractor together with $\mathcal{A}$ would contradict the extraction-error guarantee of knowledge soundness. A union bound over the polynomially many accepted proofs therefore gives

$$\Pr[\text{Bad}_{\text{ext}}] \leq \nu(\lambda).$$

It remains to compare the events used by the two hybrids. Condition on $\neg \text{Bad}_{\text{ext}}$. For each valid extracted plaintext, the value recorded by $H_{\text{sig},1}$ is

$$z_i = \Sigma.\text{Sign}(\text{sk}_{\Sigma}, m_i; \text{SKE}.\text{Enc}(k_i, \widetilde{\text{wit}}_i; r_{E,i})).$$

If $\sigma^* = z_i$, perfect correctness of the subliminal channel gives

$$\text{Extract}(\text{st}_{B,i}, \sigma^*) = \widetilde{\text{wit}}_i.$$

Since $(\text{stmt}_i, \widetilde{\text{wit}}_i) \in \text{Rel}$, this contradicts the condition defining E_{sig} . Consequently,

$$E_{\text{sig}} \subseteq E_{\text{fresh}}$$

whenever Bad_{ext} does not occur. Combining this event domination with the CRS indistinguishability and knowledge soundness gives

$$\Pr[H_{\text{sig},0} = 1] \leq \Pr[H_{\text{sig},1} = 1] + \nu(\lambda),$$

which concludes the proof. $\square$

Lemma 6.14 (Indistinguishability of $H_{\text{sig},1}$ and $H_{\text{sig},2}$). *If NIZK_{ev} is zero knowledge, then*

$$|\Pr[H_{\text{sig},1} = 1] - \Pr[H_{\text{sig},2} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by a reduction to the zero-knowledge property of NIZK_{ev} . Suppose that the output probabilities of the two hybrids differ by a non-negligible amount. We construct a zero-knowledge adversary $\mathcal{B}_{\text{zk}}$ with the same distinguishing advantage.

The reduction receives the NIZK_{ev} CRS from its challenger. It generates the NIZK_{ad} CRS with its extraction setup and answers advertisement proofs and extractions exactly as in $H_{\text{sig},1}$. On every accepted pSign query, it computes the honest evaluation statement and witness, submits them to its proving oracle, and uses the returned proof in the pre-signature. It otherwise answers the query exactly as in the hybrid. Eventually, it outputs the bit prescribed by E_{fresh} .

If the underlying challenger generates its CRS and proofs honestly, the reduction perfectly simulates $H_{\text{sig},1}$. If the challenger uses the simulation setup and simulated proofs, it perfectly simulates $H_{\text{sig},2}$. The advertisement extraction state is independent auxiliary information because the NIZK_{ad} and NIZK_{ev} CRSs are generated independently. Hence a non-negligible difference between the two hybrids contradicts zero-knowledge, proving the displayed bound. $\square$

Lemma 6.15 (Indistinguishability of $H_{\text{sig},2}$ and $H_{\text{sig},3}$). *If the FHE scheme is perfectly correct and statistically circuit private according to Definition 3.20, then*

$$|\Pr[H_{\text{sig},2} = 1] - \Pr[H_{\text{sig},3} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by an intermediate hybrid argument to the statistical circuit privacy of FHE. Order the polynomially many accepted pSign queries and define intermediate hybrids in which the first j responses are generated as in $H_{\text{sig},3}$ and all remaining responses are generated as in $H_{\text{sig},2}$. The first intermediate hybrid is $H_{\text{sig},2}$ and the last is $H_{\text{sig},3}$.

Consider the i -th replacement. The witness extracted in $H_{\text{sig},1}$ shows that $\text{pk}_{\text{FHE},i}$ is honestly generated and that $\text{ct}_{\text{wit},i}$ is an honestly supported encryption of $\widetilde{\text{wit}}_i$. The plaintext produced by the prescribed circuit is

$$z_i = \begin{cases} \Sigma.\text{Sign}(\text{sk}_\Sigma, m_i; \text{SKE}.\text{Enc}(k_i, \widetilde{\text{wit}}_i; r_{E,i})), & (\text{stmt}_i, \widetilde{\text{wit}}_i) \in \text{Rel}, \\ \perp, & (\text{stmt}_i, \widetilde{\text{wit}}_i) \notin \text{Rel}. \end{cases}$$

The extracted witness proves that $(\text{pk}_{\text{FHE},i}, \text{sk}_{\text{FHE},i})$ and $\text{ct}_{\text{wit},i}$ are in the support of $\text{FHE}.\text{KeyGen}$ and $\text{FHE}.\text{Enc}$, respectively. Since perfect correctness holds for every supported choice, including a choice made by $\mathcal{A}$, it follows that

$$\text{FHE}.\text{Dec}(\text{sk}_{\text{FHE},i}, \text{FHE}.\text{Eval}(\text{pk}_{\text{FHE},i}, C_{\theta_i}, \text{ct}_{\text{wit},i}; r_{\text{ev},i})) = z_i.$$

Hence the honestly evaluated ciphertext has the intended plaintext z_i .

Therefore, a distinguisher between two adjacent response hybrids distinguishes the two circuit-privacy distributions

$$\text{FHE}.\text{Eval}(\text{pk}_{\text{FHE},i}, C_{\theta_i}, \text{ct}_{\text{wit},i}; r_{\text{ev},i}) \quad \text{and} \quad \text{FHE}.\text{Enc}(\text{pk}_{\text{FHE},i}, z_i).$$

The evaluation proof is simulated in both adjacent hybrids and therefore does not require an opening of the fresh encryption. All preceding responses, subsequent adaptive queries, and repeated evaluations form auxiliary input to the distinguisher.

Statistical circuit privacy bounds the difference between each pair of adjacent response hybrids by $\nu(\lambda)$. Summing these differences over the polynomially many pSign queries still gives $\nu(\lambda)$, which proves the displayed bound. $\square$

Lemma 6.16 (Indistinguishability of $H_{\text{sig},3}$ and $H_{\text{sig},4}$). *If SKE has pseudorandom ciphertexts according to Definition 3.15, then*

$$|\Pr[H_{\text{sig},3} = 1] - \Pr[H_{\text{sig},4} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by an intermediate hybrid argument to the pseudorandomness of SKE ciphertexts. Order the valid pSign queries and introduce intermediate hybrids that replace the signing coins one query at a time. Suppose that two adjacent response hybrids can be distinguished with non-negligible advantage. We construct an adversary $\mathcal{B}_{\text{pr}}$ against ciphertext pseudorandomness.

For the selected query, $\mathcal{B}_{\text{pr}}$ submits $\widetilde{\text{wit}}_i$ to its encryption-or-uniform challenger and receives a value r_i^* . It computes

$$z_i = \Sigma.\text{Sign}(\text{sk}_\Sigma, m_i; r_i^*),$$

encrypts z_i under the FHE public key, and attaches a simulated evaluation proof. It generates all earlier and later responses according to the two adjacent hybrids and eventually outputs the hybrid's output bit.

If the challenger returns $r_i^* = \text{SKE.Enc}(k_i, \widetilde{\text{wit}}_i; r_{E,i})$, the reduction simulates the earlier adjacent hybrid. If it returns an independent uniform $r_i^* \leftarrow \{0, 1\}^{\ell_r}$, it simulates the later one. The key k_i is needed only for the private state $\text{st}_{B,i}$, which is not given to $\mathcal{A}$, and $\mathbf{E}_{\text{fresh}}$ refers to z_i rather than to that key or to subliminal decoding. Hence the distinguishing advantage between two adjacent hybrids is bounded by ciphertext pseudorandomness. Summing over the polynomially many valid queries preserves negligibility and proves the claim. $\square$

Lemma 6.17 (Bounding $\mathbf{H}_{\text{sig},4}$). *There is a PPT strong-unforgeability adversary $\mathcal{B}_{\text{sig}}$ such that*

$$\Pr[\mathbf{H}_{\text{sig},4} = 1] \leq \Pr[\text{SigForge}_{\mathcal{B}_{\text{sig}},\Sigma}(\lambda) = 1].$$

Proof. We prove the lemma by a direct reduction to the strong unforgeability of Σ . The reduction $\mathcal{B}_{\text{sig}}$ receives vk_{Σ} from its challenger and runs $\mathcal{A}$ using the NIZK_{ad} extraction setup and the NIZK_{ev} simulation setup. It forwards every ordinary signing query to its signing oracle.

For a **pSign** query with a valid extracted witness, the reduction queries its signing oracle on m_i and denotes the response by z_i . It returns a fresh FHE encryption of z_i and a simulated evaluation proof. For an invalid extracted witness, it instead encrypts $\perp$, simulates the proof, and makes no signing query. Since an ordinary signing-oracle response uses uniform signing coins, this perfectly simulates $\mathbf{H}_{\text{sig},4}$. The reduction does not need sk_{Σ} because the evaluation proofs are simulated.

Eventually, $\mathcal{A}$ outputs (m^*, σ^*) , which the reduction forwards to its challenger. If $\mathbf{H}_{\text{sig},4}$ outputs one, then $\Sigma.\text{Verify}(\text{vk}_{\Sigma}, m^*, \sigma^*) = 1$. Furthermore, $m^* \notin \mathcal{S}$ excludes every response to an ordinary signing query, while $\sigma^* \notin \mathcal{Z}[m^*]$ excludes every signing-oracle response used for a valid **pSign** query. Hence (m^*, σ^*) is a fresh and valid message–signature pair, even if m^* occurred in a **pSign** query. Therefore every successful outcome of $\mathbf{H}_{\text{sig},4}$ yields a successful strong-unforgeability outcome, and

$$\Pr[\mathbf{H}_{\text{sig},4} = 1] \leq \Pr[\text{SigForge}_{\mathcal{B}_{\text{sig}},\Sigma}(\lambda) = 1].$$

This concludes the proof. $\square$

6.4 Strong Canonical Signing

Lemma 6.18 (Strong canonical signing). *Suppose that NIZK_{ad} is perfectly complete, the FHE is perfectly correct, Σ is perfectly correct and supports verification-key derivation according to Definition 3.3, and SKE has pseudorandom ciphertexts. Then the construction satisfies strong canonical signing with respect to Σ according to Definition 4.7.*

Proof. To prove this lemma, we need to show that the signature obtained from the canonical adaptor execution is indistinguishable from a standard signature. In our construction, the difference between these two signatures is only the source of their signing randomness. In the canonical execution, an encryption of the witness is used, whereas in the standard signature uniform randomness is used. To show this, our proof proceeds in two steps: first, we show that, for every fixed pair $(\text{adv}_t, \text{st}_S)$ in the support of $\text{AdGen}(\text{stmt}, \text{wit})$ for a valid statement-witness pair $(\text{stmt}, \text{wit})$, the canonical execution produces a valid signature using the randomness obtained by encrypting the witness and then replaces the encryption randomness by a uniform one. The first step follows from the perfect correctness of

NIZK_{ad}, the FHE scheme, and the correctness of the signature scheme Σ . The second step directly follows from the pseudorandomness of the underlying encryption scheme.

Fix a security parameter λ , a key pair $(\text{sk}_\Sigma, \text{vk}_\Sigma)$ in the support of $\Sigma.\text{KGen}(1^\lambda)$, a message m , a valid pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, a pair $(\text{adv}_t, \text{st}_S)$ in the support of $\text{AdGen}(\text{stmt}, \text{wit})$, and a PPT distinguisher $\mathcal{D}$. Formally, we proceed through the following hybrids, each of which outputs the bit returned by $\mathcal{D}$.

Hybrid $\text{H}_{\text{can},0}$. Let

$$\sigma_{\text{can}} \leftarrow \text{Sign}^{\text{can}}(\text{sk}_\Sigma, \text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{adv}_t, \text{st}_S)$$

be the output of canonical signing according to Definition 4.7, interpreting an abort as $\perp$. In particular, let k_{SKE} and r_E denote the secret-encryption key and encryption randomness determined by the randomness sampled by pSign , and define

$$\sigma' := \Sigma.\text{Sign}(\text{sk}_\Sigma, m; \text{SKE}.\text{Enc}(k_{\text{SKE}}, \text{wit}; r_E)).$$

We define the bad-correctness event

$$\text{E}_{\text{can}} := \{\omega \mid \sigma_{\text{can}} \neq \sigma'\}.$$

The hybrid outputs

$$\mathcal{D}(\text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{adv}_t, \sigma_{\text{can}}).$$

As formalized in Lemma 6.19, the perfect correctness properties imply

$$\Pr[\text{E}_{\text{can}}] = 0.$$

Since $\Pr[\text{E}_{\text{can}}] = 0$, it holds that $\sigma_{\text{can}} = \sigma'$ with probability one. Hence $\text{H}_{\text{can},0}$ is identically distributed to the experiment that outputs

$$\mathcal{D}(\text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{adv}_t, \sigma').$$

We use this equivalent representation of $\text{H}_{\text{can},0}$ as the starting point for the next hybrid.

Hybrid $\text{H}_{\text{can},1}$. Starting from the equivalent representation above and keeping the fixed pair $(\text{adv}_t, \text{st}_S)$ unchanged, replace the value

$$\text{SKE}.\text{Enc}(k_{\text{SKE}}, \text{wit}; r_E)$$

used as the signing randomness by an independent uniform value r from the signing-randomness space. Thus, the hybrid outputs

$$\mathcal{D}(\text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{adv}_t, \Sigma.\text{Sign}(\text{sk}_\Sigma, m; r)).$$

The transition follows from ciphertext pseudorandomness of SKE , as formalized in Lemma 6.20:

$$|\Pr[\text{H}_{\text{can},0} = 1] - \Pr[\text{H}_{\text{can},1} = 1]| \leq \Pr[\text{E}_{\text{can}}] + \nu(\lambda) = \nu(\lambda).$$

The signature output in $H_{\text{can},1}$ is $\Sigma.\text{Sign}(\text{sk}_\Sigma, m; r)$ for uniform signing randomness r that is independent of the fixed advertisement, and is therefore distributed exactly as an ordinary invocation of $\Sigma.\text{Sign}(\text{sk}_\Sigma, m)$. Consequently,

$$\begin{aligned} & \left| \Pr[\mathcal{D}(\text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{advt}, \Sigma.\text{Sign}(\text{sk}_\Sigma, m)) = 1] \right. \\ & \quad \left. - \Pr[\mathcal{D}(\text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{advt}, \text{Sign}^{\text{can}}(\text{sk}_\Sigma, \text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{advt}, \text{st}_S)) = 1] \right| \\ & = |\Pr[H_{\text{can},1} = 1] - \Pr[H_{\text{can},0} = 1]| \leq \Pr[E_{\text{can}}] + \nu(\lambda) = \nu(\lambda). \end{aligned}$$

This is precisely strong canonical signing according to Definition 4.7. $\square$

Lemma 6.19 (Canonical-execution correctness). *If NIZK_{ad} is perfectly complete, the FHE is perfectly correct, and Σ is perfectly correct and supports verification-key derivation according to Definition 3.3, then, for every $\lambda \in \mathbb{N}$, every key pair $(\text{sk}_\Sigma, \text{vk}_\Sigma)$ in the support of $\Sigma.\text{KGen}(1^\lambda)$, every message m , every valid pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, and every $(\text{advt}, \text{st}_S)$ in the support of $\text{AdGen}(\text{stmt}, \text{wit})$,*

$$\Pr[E_{\text{can}}] = 0.$$

Proof. We split a possible occurrence of E_{can} according to the step of the canonical execution at which the output could differ from σ' . Consider the following events:

Event $E_{\text{can},\text{ad}}$. The advertisement proof contained in the fixed advertisement advt is rejected by pSign . Since $(\text{advt}, \text{st}_S)$ is in the support of honest advertisement generation, the statement contained in advt equals stmt and the proof is honestly generated. Thus, perfect completeness of NIZK_{ad} gives $\Pr[E_{\text{can},\text{ad}}] = 0$.

Event $E_{\text{can},\text{der}}$. The verification-key derivation check fails, that is,

$$\Sigma.\text{DerVerKey}(\text{sk}_\Sigma) \neq \text{vk}_\Sigma.$$

Since $(\text{sk}_\Sigma, \text{vk}_\Sigma)$ is in the support of $\Sigma.\text{KGen}(1^\lambda)$, correctness of verification-key derivation gives $\Pr[E_{\text{can},\text{der}}] = 0$.

Event $E_{\text{can},\text{FHE}}$. Decryption of the honestly evaluated FHE ciphertext does not yield $C_\theta(\text{wit})$. The FHE key pair, input ciphertext, and evaluated ciphertext are all honestly generated: the key pair and input ciphertext occur in the fixed supported pair $(\text{advt}, \text{st}_S)$, while pSign honestly evaluates the circuit. Hence, perfect FHE correctness gives $\Pr[E_{\text{can},\text{FHE}}] = 0$. Moreover, since $(\text{stmt}, \text{wit}) \in \text{Rel}$, the definition of C_θ gives $C_\theta(\text{wit}) = \sigma'$.

Event $E_{\text{can},\text{sig}}$. The signature σ' does not verify under vk_Σ . Perfect correctness of Σ gives $\Pr[E_{\text{can},\text{sig}}] = 0$.

Membership of $(\text{advt}, \text{st}_S)$ in the joint support of $\text{AdGen}(\text{stmt}, \text{wit})$ also ensures that st_S contains stmt and the decryption key matching the FHE public key in advt . Moreover, the fixed pair $(\text{stmt}, \text{wit})$ is valid, so the relation check in Adapt cannot fail. Consequently,

$$E_{\text{can}} \subseteq E_{\text{can},\text{ad}} \cup E_{\text{can},\text{der}} \cup E_{\text{can},\text{FHE}} \cup E_{\text{can},\text{sig}}.$$

By a union bound over the four events,

$$\Pr[E_{\text{can}}] \leq \Pr[E_{\text{can},\text{ad}}] + \Pr[E_{\text{can},\text{der}}] + \Pr[E_{\text{can},\text{FHE}}] + \Pr[E_{\text{can},\text{sig}}] = 0,$$

which proves the lemma. $\square$

Lemma 6.20 (Indistinguishability of $H_{\text{can},0}$ and $H_{\text{can},1}$). *If SKE has pseudorandom ciphertexts, then, for every PPT distinguisher $\mathcal{D}$, every $\lambda \in \mathbb{N}$, every key pair $(\text{sk}_\Sigma, \text{vk}_\Sigma)$ in the support of $\Sigma.\text{KGen}(1^\lambda)$, every message m , every valid pair $(\text{stmt}, \text{wit}) \in \text{Rel}$, and every $(\text{adv}, \text{st}_S)$ in the support of $\text{AdGen}(\text{stmt}, \text{wit})$,*

$$|\Pr[H_{\text{can},0} = 1] - \Pr[H_{\text{can},1} = 1]| \leq \Pr[E_{\text{can}}] + \nu(\lambda).$$

Proof. We prove the lemma by a reduction to the pseudorandomness of SKE. Suppose that $\mathcal{D}$ distinguishes the two hybrids with non-negligible advantage. We construct a pseudorandomness adversary $\mathcal{B}_{\text{pr}}$ with the same advantage.

First consider an intermediate experiment that outputs

$$\mathcal{D}(\text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{adv}, \sigma').$$

This experiment and $H_{\text{can},0}$ are identical unless E_{can} occurs, and hence their output probabilities differ by at most $\Pr[E_{\text{can}}]$. Due to Lemma 6.19 $H_{\text{can},0}$ and the intermediate experiment are identically distributed.

The reduction submits wit to its challenge oracle and receives a value r^* . It computes

$$\sigma^* := \Sigma.\text{Sign}(\text{sk}_\Sigma, m; r^*)$$

and outputs $\mathcal{D}(\text{vk}_\Sigma, m, \text{stmt}, \text{wit}, \text{adv}, \sigma^*)$. If the challenge oracle returns an encryption of wit under its hidden key, then r^* is distributed as $\text{SKE}.\text{Enc}(k_{\text{SKE}}, \text{wit}; r_E)$, and the reduction perfectly simulates the intermediate experiment. The fixed advertisement is independent of the fresh secret-encryption key sampled by pSign and can therefore be included unchanged in the reduction's output. If the oracle returns an independent uniform value of the same length, the reduction perfectly simulates $H_{\text{can},1}$. The difference between these two probabilities is therefore negligible by the pseudorandomness of SKE. Combining this with the probability of E_{can} gives

$$|\Pr[H_{\text{can},0} = 1] - \Pr[H_{\text{can},1} = 1]| \leq \Pr[E_{\text{can}}] + \nu(\lambda) = \nu(\lambda),$$

where the final equality again follows from Lemma 6.19. This proves the lemma. $\square$

6.5 Statement truth privacy

Lemma 6.21 (Statement truth privacy). *If the prescribed true- and false-statement distributions are computationally indistinguishable, the FHE is public-key IND-CPA secure, and NIZK_{ad} has zero knowledge, then the construction satisfies distributional statement truth privacy.*

Proof. The general idea of the proof is to gradually remove the dependence of the advertisement on the witness of the true statement. To do so, we first simulate the advertisement proof, which allows us to replace the FHE encryption of the witness by an encryption of zero using IND-CPA security. We then return to an honestly generated advertisement proof for this dummy ciphertext and replace the true statement by a statement drawn from the prescribed false-statement distribution. The resulting advertisement can be generated without a witness for the false statement and therefore defines the required statement-truth simulator.

For the formal proof, we fix a PPT distinguisher $\mathcal{D}$. We consider the statement-truth-privacy experiment, including the generation of the public parameters used to construct the advertisement, and proceed through the following hybrids. Every hybrid outputs the bit returned by $\mathcal{D}$.

Hybrid $H_{\text{tp},0}$. Sample $(\text{stmt}_T, \text{wit})$ from the prescribed true-statement distribution and generate the public parameters and advertisement honestly. This corresponds to the real distribution of Definition 4.6.

Hybrid $H_{\text{tp},1}$. Replace the NIZK_{ad} CRS by its simulation setup and replace π_{ad} by a simulated proof. The transition follows from setup indistinguishability and adaptive zero knowledge of NIZK_{ad} , as formalized in Lemma 6.22:

$$|\Pr[H_{\text{tp},0} = 1] - \Pr[H_{\text{tp},1} = 1]| \leq \nu(\lambda).$$

Hybrid $H_{\text{tp},2}$. Replace $\text{FHE}.\text{Enc}(\text{pk}_{\text{FHE}}, \text{wit})$ by $\text{FHE}.\text{Enc}(\text{pk}_{\text{FHE}}, 0^{\ell_w})$. The advertisement proof is already simulated and therefore needs neither the witness nor its encryption coins. The transition follows from IND-CPA security of FHE, as formalized in Lemma 6.23:

$$|\Pr[H_{\text{tp},1} = 1] - \Pr[H_{\text{tp},2} = 1]| \leq \nu(\lambda).$$

Hybrid $H_{\text{tp},3}$. Return to a real NIZK_{ad} CRS and replace the simulated proof by an honest proof. The key-generation coins, dummy plaintext, and dummy-encryption coins form a witness for $\mathcal{L}_{\text{ad}}$. The transition from the previous hybrid to this hybrid proceeds analogous to the transition from $H_{\text{tp},0}$ to $H_{\text{tp},1}$. Hence,

$$|\Pr[H_{\text{tp},2} = 1] - \Pr[H_{\text{tp},3} = 1]| \leq \nu(\lambda).$$

We refer to Lemma 6.22 for more details.

Hybrid $H_{\text{tp},4}$. Replace stmt_T by stmt_F from the prescribed false-statement distribution. The FHE key, dummy ciphertext, and advertisement proof are generated independently of the statement. The transition therefore follows from indistinguishability of the prescribed true- and false-statement distributions, as formalized in Lemma 6.24:

$$|\Pr[H_{\text{tp},3} = 1] - \Pr[H_{\text{tp},4} = 1]| \leq \nu(\lambda).$$

At this point, we are in the simulated world of Definition 4.6. Indeed, on input stmt_F , the simulator $\mathcal{S}$ uses the real advertisement CRS, generates an FHE key pair, encrypts 0^{ℓ_w} , and honestly proves that the key and ciphertext are well formed. It knows the required proof witness, consisting of the FHE secret key, the key-generation coins, the dummy plaintext, and the encryption coins. Since $\mathcal{L}_{\text{ad}}$ does not assert that the dummy plaintext is a witness for stmt_F , the simulator needs neither a witness for stmt_F nor a NIZK simulation trapdoor. Its output distribution is therefore exactly $H_{\text{tp},4}$.

Combining the four transitions gives

$$\begin{aligned} & \left| \Pr \left[\mathcal{D}(1^\lambda, \text{stmt}_T, \text{adv}_T) = 1 \right] - \Pr \left[\mathcal{D}(1^\lambda, \text{stmt}_F, \text{adv}_F) = 1 \right] \right| \\ & \leq \sum_{i=0}^3 |\Pr[H_{\text{tp},i} = 1] - \Pr[H_{\text{tp},i+1} = 1]| \leq \nu(\lambda), \end{aligned}$$

where adv_T is honestly generated and $\text{adv}_F \leftarrow \mathcal{S}(1^\lambda, \text{stmt}_F)$. This is precisely distributional statement truth privacy according to Definition 4.6. $\square$

6.5.1 Proofs of the Statement-Truth-Privacy Hybrid Transitions.

Lemma 6.22 (Indistinguishability of $H_{tp,0}$ and $H_{tp,1}$). *If $NIZK_{ad}$ is zero knowledge, then, for every PPT distinguisher $\mathcal{D}$,*

$$|\Pr[H_{tp,0} = 1] - \Pr[H_{tp,1} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by a reduction to the zero-knowledge property of $NIZK_{ad}$. Suppose that $\mathcal{D}$ distinguishes the two hybrids with non-negligible advantage. We construct a zero-knowledge adversary $\mathcal{B}_{zk}$ with the same advantage.

The reduction receives the $NIZK_{ad}$ CRS from its challenger and samples $(\text{stmt}_T, \text{wit})$ from the prescribed true-statement distribution. It samples ρ_K and ρ_F , generates $(\text{pk}_{FHE}, \text{sk}_{FHE})$, and computes

$$\text{ct}_{\text{wit}} = \text{FHE.Enc}(\text{pk}_{FHE}, \text{wit}; \rho_F).$$

It submits

$$x_{ad} = (\text{pk}_{FHE}, \text{ct}_{\text{wit}}) \quad \text{and} \quad w_{ad} = (\text{sk}_{FHE}, \rho_K, \text{wit}, \rho_F)$$

to its proving oracle and receives π_{ad} . It gives

$$\text{adv}_t = (\text{stmt}_T, \text{pk}_{FHE}, \text{ct}_{\text{wit}}, \pi_{ad})$$

to $\mathcal{D}$ and outputs the bit returned by $\mathcal{D}$.

If the challenger uses the real setup and proving algorithm, the reduction perfectly simulates $H_{tp,0}$. If it uses the simulation setup and simulated proof, it perfectly simulates $H_{tp,1}$. Thus a non-negligible difference between the two hybrid probabilities contradicts the assumed zero-knowledge properties. $\square$

Lemma 6.23 (Indistinguishability of $H_{tp,1}$ and $H_{tp,2}$). *If the FHE scheme is public-key IND-CPA secure, then, for every PPT distinguisher $\mathcal{D}$,*

$$|\Pr[H_{tp,1} = 1] - \Pr[H_{tp,2} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by a reduction to the IND-CPA security of the FHE scheme. Suppose that $\mathcal{D}$ distinguishes the two hybrids with non-negligible advantage. We construct an IND-CPA adversary $\mathcal{B}_{FHE}$ with the same advantage.

The reduction samples $(\text{stmt}_T, \text{wit})$ from the prescribed true-statement distribution and submits $(\text{wit}, 0^{\ell_w})$ as its challenge messages. It uses the public key and challenge ciphertext returned by its challenger as pk_{FHE} and ct_{wit} . It generates the advertisement CRS with the simulation setup and computes a simulated proof π_{ad} for $x_{ad} = (\text{pk}_{FHE}, \text{ct}_{\text{wit}})$. It then runs $\mathcal{D}$ on

$$(1^\lambda, \text{stmt}_T, (\text{stmt}_T, \text{pk}_{FHE}, \text{ct}_{\text{wit}}, \pi_{ad}))$$

and forwards its output bit to the IND-CPA challenger.

If the challenge ciphertext encrypts wit , the reduction perfectly simulates $H_{tp,1}$. If it encrypts 0^{ℓ_w} , it perfectly simulates $H_{tp,2}$. The simulated proof requires neither the FHE secret key nor an opening of the challenge ciphertext. Therefore, a non-negligible difference between the hybrid probabilities contradicts FHE IND-CPA security. $\square$

Lemma 6.24 (Indistinguishability of $H_{tp,3}$ and $H_{tp,4}$). *If the prescribed true- and false-statement distributions are computationally indistinguishable, then, for every PPT distinguisher $\mathcal{D}$,*

$$|\Pr[H_{tp,3} = 1] - \Pr[H_{tp,4} = 1]| \leq \nu(\lambda).$$

Proof. We prove the lemma by a reduction to the indistinguishability of the prescribed statement distributions. Suppose that $\mathcal{D}$ distinguishes the two hybrids with non-negligible advantage. We construct a distinguisher $\mathcal{B}_{\text{stmt}}$ for the statement distributions.

On receiving a statement stmt from its challenger, the reduction generates the real NIZK_{ad} CRS, an FHE key pair, and an encryption ct_0 of 0^{ℓ_w} . Since it knows the key-generation and encryption coins, it generates an honest proof for $x_{\text{ad}} = (\text{pk}_{\text{FHE}}, \text{ct}_0)$. It gives

$$(1^\lambda, \text{stmt}, (\text{stmt}, \text{pk}_{\text{FHE}}, \text{ct}_0, \pi_{\text{ad}}))$$

to $\mathcal{D}$ and outputs the resulting bit.

If stmt is drawn from the true-statement marginal, this perfectly simulates $H_{\text{tp},3}$. If it is drawn from the false-statement distribution, it perfectly simulates $H_{\text{tp},4}$. All advertisement components other than stmt are generated independently of its truth value. Thus a non-negligible difference between the hybrid probabilities contradicts the assumed indistinguishability of the statement distributions. $\square$

Acknowledgements

The authors used OpenAI Codex (GPT-5.6) and Claude, to assist with drafting and editing the manuscript, structuring technical arguments, and resolving LaTeX issues. All AI-assisted content was reviewed, revised, and independently validated by the authors, who take full responsibility for the final manuscript.

Nico Döttling and Riccardo Zanotto: Funded by the European Union (ERC, LACONIC, 101041207). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. Sri AravindaKrishnan Thyagarajan was supported by a gift from SUI Academic grants.

References

- [1] Asharov, G., Jain, A., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. Cryptology ePrint Archive, Report 2011/613 (2011), <https://eprint.iacr.org/2011/613>
- [2] Aumayr, L., Ersoy, O., Erwig, A., Faust, S., Hostáková, K., Maffei, M., Moreno-Sanchez, P., Riahi, S.: Generalized channels from limited blockchain scripts and adaptor signatures. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part II. LNCS, vol. 13091, pp. 635–664. Springer, Cham (Dec 2021). doi:[10.1007/978-3-030-92075-3_22](https://doi.org/10.1007/978-3-030-92075-3_22)
- [3] Bartoli, C., Cascudo, I.: On sigma-protocols and (packed) black-box secret sharing schemes. In: Tang, Q., Teague, V. (eds.) PKC 2024, Part II. LNCS, vol. 14602, pp. 426–457. Springer, Cham (Apr 2024). doi:[10.1007/978-3-031-57722-2_14](https://doi.org/10.1007/978-3-031-57722-2_14)
- [4] Bellare, M., Rogaway, P.: PSS: Provably secure encoding method for digital signatures. In: IEEE P1363a: Provably Secure Signatures. pp. 1–14. IEEE Computer Society Press, Los Alamitos, CA, USA (1998)

- [5] Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Everything provable is provable in zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 37–56. Springer, New York (Aug 1990). doi:[10.1007/0-387-34799-2_4](https://doi.org/10.1007/0-387-34799-2_4)
- [6] Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Berlin, Heidelberg (Feb 2005). doi:[10.1007/978-3-540-30576-7_18](https://doi.org/10.1007/978-3-540-30576-7_18)
- [7] Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational diffie-hellman. In: Proceedings of the 9th International Conference on Theory and Practice of Public-Key Cryptography. p. 229–240. PKC'06, Springer-Verlag, Berlin, Heidelberg (2006). doi:[10.1007/11745853_15](https://doi.org/10.1007/11745853_15), https://doi.org/10.1007/11745853_15
- [8] Bourse, F., Pino, R.D., Minelli, M., Wee, H.: FHE circuit privacy almost for free. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14–18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 62–89. Springer (2016). doi:[10.1007/978-3-662-53008-5_3](https://doi.org/10.1007/978-3-662-53008-5_3), https://doi.org/10.1007/978-3-662-53008-5_3
- [9] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. Cryptology ePrint Archive, Paper 2017/1066 (2017), <https://eprint.iacr.org/2017/1066>
- [10] Campanelli, M., Gennaro, R., Goldfeder, S., Nizzardo, L.: Zero-knowledge contingent payments revisited: Attacks and payments for services. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. p. 229–243. CCS '17, Association for Computing Machinery, New York, NY, USA (2017). doi:[10.1145/3133956.3134060](https://doi.org/10.1145/3133956.3134060), <https://doi.org/10.1145/3133956.3134060>
- [11] Cartor, R., Daly, N., Gaggero, G., LeGrow, J.T., Sanguineti, A., Sconza, S.: Post-quantum adaptor signatures with strong security from cryptographic group actions. In: Bardet, M., Niederhagen, R. (eds.) Post-Quantum Cryptography - 17th International Workshop, PQCrypto 2026, Saint-Malo, France, April 14–16, 2026, Proceedings, Part I. Lecture Notes in Computer Science, vol. 16491, pp. 370–405. Springer (2026). doi:[10.1007/978-3-032-22695-2_13](https://doi.org/10.1007/978-3-032-22695-2_13), https://doi.org/10.1007/978-3-032-22695-2_13
- [12] Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable proof systems and applications. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 281–300. Springer, Berlin, Heidelberg (Apr 2012). doi:[10.1007/978-3-642-29011-4_18](https://doi.org/10.1007/978-3-642-29011-4_18)
- [13] Chase, M., Orrù, M., Perrin, T., Zaverucha, G.: Proofs of discrete logarithm equality across groups. Cryptology ePrint Archive, Report 2022/1593 (2022), <https://eprint.iacr.org/2022/1593>
- [14] Ciampi, M., Liu, X., Tzannetos, I., Zikas, V.: Universal adaptor signatures from blackbox multi-party computation. In: Topics in Cryptology – CT-RSA 2025: Cryptographers' Track at the RSA Conference 2025, San Francisco, CA, USA, April 28–May 1, 2025, Proceedings. p. 375–398. Springer-Verlag, Berlin, Heidelberg (2025). doi:[10.1007/978-3-031-88661-4_16](https://doi.org/10.1007/978-3-031-88661-4_16), https://doi.org/10.1007/978-3-031-88661-4_16

- [15] Cleve, R.: Limits on the security of coin flips when half the processors are faulty. In: Proceedings of the eighteenth annual ACM symposium on Theory of computing. pp. 364–369. ACM, New York, NY, USA (1986). doi:[10.1145/12130.12168](https://doi.org/10.1145/12130.12168), <https://doi.org/10.1145/12130.12168>
- [16] Dai, W., Okamoto, T., Yamamoto, G.: Stronger security and generic constructions for adaptor signatures. In: Isobe, T., Sarkar, S. (eds.) Progress in Cryptology – INDOCRYPT 2022. pp. 52–77. Springer International Publishing, Cham (2022)
- [17] Damgård, I., Mikkelsen, G.L.: Efficient, robust and constant-round distributed RSA key generation. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 183–200. Springer, Berlin, Heidelberg (Feb 2010). doi:[10.1007/978-3-642-11799-2_12](https://doi.org/10.1007/978-3-642-11799-2_12)
- [18] Decker, C., Wattenhofer, R.: A fast and scalable payment network with bitcoin duplex micro-payment channels. In: Pelc, A., Schwarzmann, A.A. (eds.) Stabilization, Safety, and Security of Distributed Systems. pp. 3–18. Springer International Publishing, Cham (2015)
- [19] Della Monica, P., Visconti, I., Vitaletti, A., Zecchini, M.: Decentralized fair exchange with advertising. In: Cryptology and Network Security: 24th International Conference, CANS 2025, Osaka, Japan, November 17–20, 2025, Proceedings. p. 358–381. Springer-Verlag, Berlin, Heidelberg (2025). doi:[10.1007/978-981-95-4434-9_17](https://doi.org/10.1007/978-981-95-4434-9_17), https://doi.org/10.1007/978-981-95-4434-9_17
- [20] Döttling, N., Mohamed, M., Zanotto, R.: GATOR: group action adaptor signatures via mpc-in-the-head. IACR Cryptol. ePrint Arch. **2026**, 1067 (2026), <https://eprint.iacr.org/2026/1067>
- [21] Ducas, L., Stehlé, D.: Sanitization of FHE ciphertexts. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 294–310. Springer, Berlin, Heidelberg (May 2016). doi:[10.1007/978-3-662-49890-3_12](https://doi.org/10.1007/978-3-662-49890-3_12)
- [22] Dziembowski, S., ECKEY, L., Faust, S.: FairSwap: How to fairly exchange digital goods. Cryptology ePrint Archive, Paper 2018/740 (2018). doi:[10.1145/3243734.3243857](https://doi.org/10.1145/3243734.3243857), <https://eprint.iacr.org/2018/740>
- [23] Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory **31**(4), 469–472 (1985). doi:[10.1109/TIT.1985.1057074](https://doi.org/10.1109/TIT.1985.1057074)
- [24] Erwig, A., Faust, S., Hostáková, K., Maitra, M., Riahi, S.: Two-party adaptor signatures from identification schemes. In: Garay, J. (ed.) PKC 2021, Part I. LNCS, vol. 12710, pp. 451–480. Springer, Cham (May 2021). doi:[10.1007/978-3-030-75245-3_17](https://doi.org/10.1007/978-3-030-75245-3_17)
- [25] Esgin, M.F., Ersoy, O., Erkin, Z.: Post-quantum adaptor signatures and payment channel networks. Cryptology ePrint Archive, Paper 2020/845 (2020), <https://eprint.iacr.org/2020/845>
- [26] Fortnow, L.: The complexity of perfect zero-knowledge (extended abstract). In: Aho, A. (ed.) 19th ACM STOC. pp. 204–209. ACM Press (May 1987). doi:[10.1145/28395.28418](https://doi.org/10.1145/28395.28418)
- [27] Fuchsbauer, G.: WI is not enough: Zero-knowledge contingent (service) payments revisited. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 49–62. ACM Press (Nov 2019). doi:[10.1145/3319535.3354234](https://doi.org/10.1145/3319535.3354234)

- [28] Garg, S., Gentry, C., Sahai, A., Waters, B.: Witness encryption and its applications. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 467–476. ACM Press (Jun 2013). doi:[10.1145/2488608.2488667](https://doi.org/10.1145/2488608.2488667)
- [29] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 169–178. ACM Press (May / Jun 2009). doi:[10.1145/1536414.1536440](https://doi.org/10.1145/1536414.1536440)
- [30] Gerhart, P., Schröder, D., Soni, P., Thyagarajan, S.A.K.: Foundations of adaptor signatures. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part II. LNCS, vol. 14652, pp. 161–189. Springer, Cham (May 2024). doi:[10.1007/978-3-031-58723-8_6](https://doi.org/10.1007/978-3-031-58723-8_6)
- [31] Glaeser, N., Maffei, M., Malavolta, G., Moreno-Sanchez, P., Tairi, E., Thyagarajan, S.A.K.: Foundations of coin mixing services. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 1259–1273. ACM Press (Nov 2022). doi:[10.1145/3548606.3560637](https://doi.org/10.1145/3548606.3560637)
- [32] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC. pp. 218–229. ACM Press (May 1987). doi:[10.1145/28395.28420](https://doi.org/10.1145/28395.28420)
- [33] Gong, B., Lau, W.F., Au, M.H., Yang, R., Xue, H., Li, L.: Efficient zero-knowledge arguments for paillier cryptosystem. In: 2024 IEEE Symposium on Security and Privacy. pp. 1813–1831. IEEE Computer Society Press (May 2024). doi:[10.1109/SP54263.2024.00093](https://doi.org/10.1109/SP54263.2024.00093)
- [34] Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Berlin, Heidelberg (May / Jun 2006). doi:[10.1007/11761679_21](https://doi.org/10.1007/11761679_21)
- [35] Halevi, S., Krawczyk, H.: Strengthening digital signatures via randomized hashing. In: Dwork, C. (ed.) Advances in Cryptology - CRYPTO 2006. pp. 41–59. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
- [36] Hanzlik, L., Loss, J., Thyagarajan, S.A.K., Wagner, B.: Sweep-uc: Swapping coins privately. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 3822–3839. IEEE Computer Society, Los Alamitos, CA, USA (may 2024). doi:[10.1109/SP54263.2024.00081](https://doi.org/10.1109/SP54263.2024.00081), <https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00081>
- [37] Kutyłowski, K., Persiano, G., Phan, D.H., Yung, M., Zawada, D.: Anamorphic signatures: Secrecy from a dictator who only permits authentication! In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2023. Lecture Notes in Computer Science, vol. 14148, pp. 732–762. Springer (2023). doi:[10.1007/978-3-031-38545-2_25](https://doi.org/10.1007/978-3-031-38545-2_25)
- [38] Liu, X., Tzannetos, I., Zikas, V.: Adaptor signatures: New security definition and a generic construction for NP relations. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part II. LNCS, vol. 15485, pp. 168–193. Springer, Singapore (Dec 2024). doi:[10.1007/978-981-96-0888-1_6](https://doi.org/10.1007/978-981-96-0888-1_6)
- [39] Madathil, V., Thyagarajan, S.A.K., Vasilopoulos, D., Fournier, L., Malavolta, G., Moreno-Sanchez, P.: Cryptographic oracle-based conditional payments (01 2023). doi:[10.14722/ndss.2023.24024](https://doi.org/10.14722/ndss.2023.24024)

- [40] Maxwell, G.: Zero knowledge contingent payment. Bitcoin Wiki (2011)
- [41] Miller, A., Bentov, I., Bakshi, S., Kumaresan, R., McCorry, P.: Sprites and state channels: Payment networks that go faster than lightning. In: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, vol. 11598, pp. 508–526. Springer, Cham (Feb 2019). doi:[10.1007/978-3-030-32101-7_30](https://doi.org/10.1007/978-3-030-32101-7_30)
- [42] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (May 2009), <http://www.bitcoin.org/bitcoin.pdf>
- [43] Nguyen, K., Ambrona, M., Abe, M.: WI is almost enough: Contingent payment all over again. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020. pp. 641–656. ACM Press (Nov 2020). doi:[10.1145/3372297.3417888](https://doi.org/10.1145/3372297.3417888)
- [44] Poelstra, A.: Scriptless scripts. Presentation Slides (2017)
- [45] Simmons, G.J.: The subliminal channel and digital signatures. In: Advances in Cryptology: Proceedings of CRYPTO '83. Lecture Notes in Computer Science, vol. 196, pp. 364–378. Springer (1985). doi:[10.1007/3-540-39568-7_28](https://doi.org/10.1007/3-540-39568-7_28)
- [46] Simmons, G.J.: Subliminal communication is easy using the dsa. In: Hellese, T. (ed.) Advances in Cryptology — EUROCRYPT '93. pp. 218–232. Springer Berlin Heidelberg, Berlin, Heidelberg (1994)
- [47] Tairi, E., Moreno-Sanchez, P., Maffei, M.: Post-quantum adaptor signature for privacy-preserving off-chain payments. Cryptology ePrint Archive, Paper 2020/1345 (2020), <https://eprint.iacr.org/2020/1345>
- [48] Teutsch, J., Reitwießner, C.: A scalable verification solution for blockchains. arXiv preprint arXiv:1908.04756 (2019). doi:[10.48550/arXiv.1908.04756](https://doi.org/10.48550/arXiv.1908.04756), <https://arxiv.org/pdf/1908.04756>, preprint
- [49] Thyagarajan, S.A.K., Malavolta, G., Moreno-Sanchez, P.: Universal atomic swaps: Secure exchange of coins across all blockchains. In: 2022 IEEE Symposium on Security and Privacy. pp. 1299–1316. IEEE Computer Society Press (May 2022). doi:[10.1109/SP46214.2022.9833731](https://doi.org/10.1109/SP46214.2022.9833731)
- [50] Vanjani, N., Greiner, G., Thyagarajan, S.A., Soni, P.: New constructions of functional adaptor signatures: Broader functions and improved efficiency. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P). IEEE (2026), to appear
- [51] Vanjani, N., Soni, P., Thyagarajan, S.A.K.: Functional adaptor signatures: Beyond all-or-nothing blockchain-based payments. Cryptology ePrint Archive, Paper 2024/1523 (2024). doi:[10.1145/3658644.3690240](https://doi.org/10.1145/3658644.3690240), <https://eprint.iacr.org/2024/1523>

A A Sigma Protocol for $\mathcal{L}_{\text{CE}}$

In this section we describe a Σ protocol for the ciphertext equality language $\mathcal{L}_{\text{CE}}$ defined in Equation (1), which we recall here.

$$\mathcal{L}_{\text{CE}} = \left\{ \left(\text{pk}_{\text{LE}}, \text{pk}_{\text{HE}}, \{c_{\text{LE},i}\}_{i \in [\ell]}, \{c_{\text{HE},i}\}_{i \in [\ell]}; \{m_i\}_{i \in [\ell]}, \{r_{\text{LE},i}\}_{i \in [\ell]}, \{r_{\text{HE},i}\}_{i \in [\ell]} \right) : m_i \in \{0, 1\}, \right. \\ \left. c_{\text{LE},i} = \text{LE.Enc}(\text{pk}_{\text{LE}}, m_i; r_{\text{LE},i}) \text{ and } c_{\text{HE},i} = \text{HE.Enc}(\text{pk}_{\text{HE}}, m_i; r_{\text{HE},i}) \text{ for all } i \in [\ell] \right\}$$

We rely on a CRS which consists of the public key of another homomorphic encryption scheme pk_{crs} that satisfies the two properties of **Add** and **Mul** of **LE** and **HE**; we denote it by $(\text{KGen}, \text{Enc}, \text{Dec})$. In particular, it can be instantiated by **LE** itself. The proof additionally contains, for every $i \in [\ell]$, a proof $\pi_{\text{bit},i}$ from a sound **NIZK** **NIZK**_{bit} that $c_{\text{HE},i}$ encrypts a bit.

The protocol works for groups of unknown order. Let

$$\mathcal{C} := \{0, \dots, C-1\}, \quad \mathcal{U} := \{0, \dots, U-1\}, \quad \mathcal{R}_{\text{mask}} := \{0, \dots, R-1\}$$

be, respectively, the challenge space, plaintext-mask space, and randomness-mask space. Let B be a public upper bound on the integer coins in the encryption randomness spaces, and let Q be a public lower bound on the effective plaintext orders of the three encryption schemes. We require

$$U + C < Q, \quad \frac{1}{C} = \nu(\lambda), \quad \frac{C}{U} = \nu(\lambda), \quad \text{and} \quad \frac{CB}{R} = \nu(\lambda). \quad (2)$$

Only the bounds B and Q , and not the group orders themselves, are needed. The randomness spaces of the three encryption schemes are represented by integer coins in $[0, B)$; thus, the randomness values occurring in a valid witness already have such a representation, and the prover never needs to reduce them modulo an unknown order. In the concrete instantiation, the **LE** and **CRS** plaintext groups have public prime order, while the well-formedness proof for the **BGN** key also certifies the required lower bound on its effective plaintext order. We require efficient public-key and ciphertext validity checks, and that every ciphertext passing these checks has an opening to its decrypted plaintext. The verifier rejects if a validity check fails. These properties hold for the group-based instantiations after the usual group-membership checks. All masks and responses below are integers and are never reduced modulo a group order; reduction happens only implicitly when the encryption algorithms perform group operations.

We state the protocol below.

Sigma protocol for $\mathcal{L}_{\text{CE}}$

We describe an interactive proof for $\mathcal{L}_{\text{CE}}$ in commitment–challenge–response form.^a We define the **Prove**₁, **Prove**₂, **Chal**, and **Verify** algorithms. As mentioned above, our protocol works for encryption schemes satisfying the **Add** and **Mul** relations. For our application, we instantiate them concretely using **ElGamal** for **LE** and the extractable encryption, and **BGN** for **HE**.

Prove₁(crs, stmt, wit)

1. Parse $\text{crs} = (\text{pk}_{\text{crs}}, \text{crs}_{\text{bit}})$, $\text{stmt} = (\text{pk}_{\text{HE}}, \text{pk}_{\text{LE}}, \{\text{ct}_{\text{LE},j}\}_{j \in [\ell]}, \{\text{ct}_{\text{HE},j}\}_{j \in [\ell]})$, and $\text{wit} = (\{m_j\}_{j \in [\ell]}, \{r_{\text{HE},j}\}_{j \in [\ell]}, \{r_{\text{LE},j}\}_{j \in [\ell]})$.
2. Sample $u_j \leftarrow \$ \mathcal{U}$ and integer randomness masks $r'_{\text{crs},j}, r'_{\text{HE},j}, r'_{\text{LE},j} \leftarrow \$ \mathcal{R}_{\text{mask}}$ for all $j \in [\ell]$.

3. For all $j \in [\ell]$, generate:

- $\text{ct}_{\text{crs},j} := \text{Enc}(\text{pk}_{\text{crs}}, m_j; r_{\text{crs},j})$, $\text{ct}'_{\text{crs},j} := \text{Enc}(\text{pk}_{\text{crs}}, u_j; r'_{\text{crs},j})$,
- $\text{ct}'_{\text{HE},j} := \text{HE.Enc}(\text{pk}_{\text{HE}}, u_j; r'_{\text{HE},j})$, and
- $\text{ct}'_{\text{LE},j} := \text{LE.Enc}(\text{pk}_{\text{LE}}, u_j; r'_{\text{LE},j})$ and
 $\pi_{\text{bit},j} \leftarrow \text{NIZK}_{\text{bit}}.\text{Prove}(\text{crs}_{\text{bit}}, (\text{pk}_{\text{HE}}, \text{ct}_{\text{HE},j}), (m_j, r_{\text{HE},j})),$

where $r_{\text{crs},j}$ is sampled from the randomness space of the CRS encryption scheme as an integer smaller than B .

4. Output $\text{com} := (\text{ct}_{\text{crs},j}, \text{ct}'_{\text{crs},j}, \text{ct}'_{\text{HE},j}, \text{ct}'_{\text{LE},j}, \pi_{\text{bit},j})_{j \in [\ell]}$ and
 $\text{st} = (m_j, r_{\text{HE},j}, r_{\text{LE},j}, u_j, r_{\text{crs},j}, r'_{\text{crs},j}, r'_{\text{HE},j}, r'_{\text{LE},j})_{j \in [\ell]}.$

$\text{Chal}(\text{crs}, \text{stmt}, \text{com})$

1. Output $c_j \leftarrow \mathcal{C}$ for all $j \in [\ell]$.

$\text{Prove}_2(\text{st}, \{c_j\}_{j \in [\ell]})$

1. Parse $\text{st} = (m_j, r_{\text{HE},j}, r_{\text{LE},j}, u_j, r_{\text{crs},j}, r'_{\text{crs},j}, r'_{\text{HE},j}, r'_{\text{LE},j})_{j \in [\ell]}.$
2. For all $j \in [\ell]$, compute over the integers

$$\begin{aligned} z_j &:= u_j + c_j m_j, & r''_{\text{crs},j} &:= r'_{\text{crs},j} + c_j r_{\text{crs},j}, \\ r''_{\text{HE},j} &:= r'_{\text{HE},j} + c_j r_{\text{HE},j}, & r''_{\text{LE},j} &:= r'_{\text{LE},j} + c_j r_{\text{LE},j}. \end{aligned}$$

3. Output $\pi := (z_j, r''_{\text{crs},j}, r''_{\text{HE},j}, r''_{\text{LE},j})_{j \in [\ell]}.$

$\text{Verify}(\text{crs}, \text{com}, \pi)$

1. Parse $\text{crs} = (\text{pk}_{\text{crs}}, \text{crs}_{\text{bit}})$, $\text{com} := (\text{ct}_{\text{crs},j}, \text{ct}'_{\text{crs},j}, \text{ct}'_{\text{HE},j}, \text{ct}'_{\text{LE},j}, \pi_{\text{bit},j})_{j \in [\ell]}$, and
 $\pi := (z_j, r''_{\text{crs},j}, r''_{\text{HE},j}, r''_{\text{LE},j})_{j \in [\ell]}.$
2. Check for all $j \in [\ell]$ if:
 - the public keys and all ciphertexts pass their validity and group-membership checks,
 - $\text{NIZK}_{\text{bit}}.\text{Verify}(\text{crs}_{\text{bit}}, (\text{pk}_{\text{HE}}, \text{ct}_{\text{HE},j}), \pi_{\text{bit},j}) = 1$,
 - $0 \leq z_j < U + C$ and $0 \leq r''_{X,j} < R + CB$ for $X \in \{\text{crs}, \text{HE}, \text{LE}\}$,
 - $\text{Add}(\text{ct}'_{\text{crs},j}, \text{Mul}(c_j, \text{ct}_{\text{crs},j})) \stackrel{?}{=} \text{Enc}(\text{pk}_{\text{crs}}, z_j; r''_{\text{crs},j})$,
 - $\text{HE.Add}(\text{ct}'_{\text{HE},j}, \text{HE.Mul}(c_j, \text{ct}_{\text{HE},j})) \stackrel{?}{=} \text{HE.Enc}(\text{pk}_{\text{HE}}, z_j; r''_{\text{HE},j})$, and

- $\text{LE.Add}(\text{ct}'_{\text{LE},j}, \text{LE.Mul}(c_j, \text{ct}_{\text{LE},j})) \stackrel{?}{=} \text{LE.Enc}(\text{pk}_{\text{LE}}, z_j; r''_{\text{LE},j})$.

3. If the above is true, output 1, otherwise, output 0.

^aThe extractor recovers only the message component of the witness. We therefore prove the required partial extraction property directly, instead of relying on special soundness.

We now prove completeness, partial knowledge soundness, and honest-verifier zero-knowledge.

Lemma A.1 (Completeness). *If $\text{Rel}_{\text{CE}}(\text{stmt}, \text{wit}) = 1$ and the prover and verifier follow the protocol, then the verifier accepts with probability 1.*

Proof. The three verification equations follow directly from the homomorphic properties of **Add** and **Mul**. Moreover, since $m_j \in \{0, 1\}$, we have $0 \leq z_j = u_j + c_j m_j < U + C$, and since the input randomness values are smaller than B , we have $0 \leq r''_{X,j} < R + CB$. Completeness of NIZK_{bit} gives the remaining check. $\square$

Lemma A.2 (Partial Knowledge Soundness). *Let ϵ_{bit} be the soundness error of one bit proof. The above protocol has a straight-line extractor for the message component of the witness. Except with probability at most $\ell \epsilon_{\text{bit}} + 2\ell/C$, its output on an accepting transcript is a bit vector that can be extended by encryption randomness to a witness for $\mathcal{L}_{\text{CE}}$.*

Proof. We first define the extractor.

Extractor

$\text{Ext}_1(1^\lambda)$:

1. Generate $(\text{pk}_{\text{crs}}, \text{sk}_{\text{crs}}) \leftarrow \text{KGen}(1^\lambda)$ and $\text{crs}_{\text{bit}} \leftarrow \text{NIZK}_{\text{bit}}.\text{Setup}(1^\lambda)$.
2. Output $\text{crs} := (\text{pk}_{\text{crs}}, \text{crs}_{\text{bit}})$ and $\text{st} := \text{sk}_{\text{crs}}$.

$\text{Ext}_2(\text{st}, \text{com}, \{c_j\}_{j \in [\ell]}, \pi)$:

1. On input an accepting transcript $\text{com} := (\text{ct}_{\text{crs},j}, \text{ct}'_{\text{crs},j}, \text{ct}'_{\text{HE},j}, \text{ct}'_{\text{LE},j}, \pi_{\text{bit},j})_{j \in [\ell]}$, $\{c_j\}_{j \in [\ell]}$, and $\pi := (z_j, r''_{\text{crs},j}, r''_{\text{HE},j}, r''_{\text{LE},j})_{j \in [\ell]}$, compute $\hat{m}_j \leftarrow \text{Dec}(\text{sk}_{\text{crs}}, \text{ct}_{\text{crs},j})$ for every j . If every $\hat{m}_j \in \{0, 1\}$, output $\hat{m} = (\hat{m}_1, \dots, \hat{m}_\ell)$; otherwise output $\perp$.

We call this a *partial extractor*: the witness for $\mathcal{L}_{\text{CE}}$ also contains the encryption randomness, but the extractor returns only its message projection. This is sufficient for the compiler in Section 5. The CRS output by Ext_1 is identically distributed to an honest CRS: pk_{crs} is generated by **KGen**, and crs_{bit} is generated by the honest setup of NIZK_{bit} .

It remains to prove the claimed extraction guarantee. Fix an arbitrary malicious first-stage prover and condition on its output commitment. All ciphertexts, and hence all their plaintexts, are now fixed before the verifier samples the challenge vector. By a union bound, the probability that one of the ℓ accepted bit proofs is false is at most $\ell \epsilon_{\text{bit}}$. We condition first on the complementary event, so every $\text{ct}_{\text{HE},j}$ has a plaintext $b_j \in \{0, 1\}$.

Fix a coordinate j and suppress the index. For $X \in \{\text{crs}, \text{HE}, \text{LE}\}$, let q_X be the effective plaintext order and write

$$\text{ct}_X = \text{Enc}_X(\text{pk}_X, m_X; \rho_X), \quad \text{ct}'_X = \text{Enc}_X(\text{pk}_X, a_X; \rho'_X),$$

using canonical plaintext representatives $m_X, a_X \in \{0, \dots, q_X - 1\}$. Here $m_{\text{HE}} = b$, while $m_{\text{crs}} = \widehat{m}$ is the value obtained by the decryption performed in Ext_2 ; the extractor returns it only if it is a bit. Ciphertext validity guarantees the existence of the displayed openings; the extractor need not know them.

The three ciphertext equations checked by the verifier are, uniformly for $X \in \{\text{crs}, \text{HE}, \text{LE}\}$,

$$\text{Add}_X(\text{ct}'_X, \text{Mul}_X(c, \text{ct}_X)) = \text{Enc}_X(\text{pk}_X, z; r''_X). \quad (3)$$

By the homomorphic relations, the left-hand side decrypts to $a_X + cm_X$ modulo q_X , whereas the right-hand side decrypts to z modulo q_X . Thus every accepting transcript satisfies

$$\begin{aligned} a_{\text{crs}} + c\widehat{m} &\equiv z \pmod{q_{\text{crs}}}, \\ a_{\text{HE}} + cb &\equiv z \pmod{q_{\text{HE}}}, \\ a_{\text{LE}} + cm_{\text{LE}} &\equiv z \pmod{q_{\text{LE}}}. \end{aligned} \quad (4)$$

Moreover, the verifier checks $0 \leq z < U + C < Q \leq q_X$. Therefore z is already the canonical representative in every plaintext group, and Equation (4) equivalently gives

$$z = [a_{\text{crs}} + c\widehat{m}]_{q_{\text{crs}}} = [a_{\text{HE}} + cb]_{q_{\text{HE}}} = [a_{\text{LE}} + cm_{\text{LE}}]_{q_{\text{LE}}}, \quad (5)$$

where $[\cdot]_q$ denotes the canonical representative modulo q .

Suppose that $m_X \neq b$ for some $X \in \{\text{crs}, \text{LE}\}$; this includes the case in which $\widehat{m} \notin \{0, 1\}$ and Ext_2 returns $\perp$. We bound the challenges $c \in \mathcal{C}$ for which Equation (5) can hold. If $b = 0$, then $[a_{\text{HE}} + cb]_{q_{\text{HE}}} = a_{\text{HE}}$ is constant. Hence acceptance requires

$$a_X + cm_X \equiv a_{\text{HE}} \pmod{q_X}.$$

Since $m_X \neq 0$, q_X is prime, and $C < q_X$, this equation has at most one solution $c \in \mathcal{C}$.

If $b = 1$, the function $c \mapsto [a_{\text{HE}} + c]_{q_{\text{HE}}}$ has at most two affine pieces on $\mathcal{C}$, because $C < q_{\text{HE}}$ and hence $a_{\text{HE}} + c$ can cross a multiple of q_{HE} at most once. On a fixed piece there is a constant $k \in \{0, 1\}$ such that acceptance implies

$$a_X + cm_X \equiv a_{\text{HE}} + c - kq_{\text{HE}} \pmod{q_X},$$

or equivalently

$$c(m_X - 1) \equiv a_{\text{HE}} - kq_{\text{HE}} - a_X \pmod{q_X}. \quad (6)$$

Because $m_X \neq 1$ and q_X is prime, the coefficient $m_X - 1$ is invertible modulo q_X . Equation (6) therefore has a unique solution modulo q_X , and at most one solution in $\mathcal{C}$ since $C < q_X$. There are at most two pieces, so there are at most two accepting challenges in total.

The challenge c_j is uniform in $\mathcal{C}$ and independent of the commitment. Thus, for each coordinate at which either $m_{\text{crs}} \neq b$ or $m_{\text{LE}} \neq b$, acceptance has probability at most $2/C$. A union bound over the ℓ coordinates, together with the bit-proof soundness error, gives

$$\Pr[\text{Verify} = 1 \wedge (\widehat{m} \text{ is not a common bit plaintext})] \leq \ell\epsilon_{\text{bit}} + \frac{2\ell}{C}.$$

Consequently, outside this event, $\hat{m}_j = b_j = m_{\text{LE},j}$ for all j . By ciphertext validity, there exist randomness values opening both input ciphertext vectors to $\hat{m}$, so it can be extended to a witness for $\mathcal{L}_{\text{CE}}$; the extractor neither computes nor returns those values.

The extractor is straight-line and can therefore be applied independently to every adaptive proof received by the compiler. After Fiat–Shamir, a union bound over the adversary’s polynomially many proof and random-oracle queries preserves negligibility by Equation (2). More generally, the same argument applies to non-prime q_{crs} or q_{LE} whenever multiplication by every nonzero plaintext difference is injective over $\mathcal{C}$. $\square$

Lemma A.3 (Computational Honest-Verifier Zero-Knowledge). *The protocol satisfies computational honest-verifier zero-knowledge. Namely, for every statement in $\mathcal{L}_{\text{CE}}$ and every challenge vector $\{c_j\}_{j \in [\ell]} \in \mathcal{C}^\ell$, there is a PPT simulator whose joint output $(\text{crs}, \text{transcript})$ is computationally indistinguishable from an honestly generated CRS and transcript with that challenge vector.*

Proof. Recall that subtraction is implemented homomorphically as $\text{Subt}(\text{ct}_1, \text{ct}_2) = \text{Add}(\text{ct}_1, \text{Mul}(-1, \text{ct}_2))$, and analogously for HE and LE. We first describe the simulator and then prove indistinguishability through hybrids.

The simulator generates an honest CRS-encryption key pk_{crs} and runs the CRS-generation algorithm of the zero-knowledge simulator for NIZK_{bit} , obtaining $(\text{crs}_{\text{bit}}, \tau_{\text{bit}})$. On input a statement and a fixed challenge vector $\{c_j\}_{j \in [\ell]}$, it then samples, for every $j \in [\ell]$,

$$z_j \leftarrow \$\mathcal{U}, \quad r''_{\text{crs},j}, r''_{\text{HE},j}, r''_{\text{LE},j} \leftarrow \$\mathcal{R}_{\text{mask}},$$

generates $\text{ct}_{\text{crs},j} \leftarrow \text{Enc}(\text{pk}_{\text{crs}}, 0)$, and uses τ_{bit} to obtain a simulated bit proof $\pi_{\text{bit},j}$ for $(\text{pk}_{\text{HE}}, \text{ct}_{\text{HE},j})$. It sets

$$\begin{aligned} \text{ct}'_{\text{crs},j} &:= \text{Subt}(\text{Enc}(\text{pk}_{\text{crs}}, z_j; r''_{\text{crs},j}), \text{Mul}(c_j, \text{ct}_{\text{crs},j})), \\ \text{ct}'_{\text{HE},j} &:= \text{HE.Subt}(\text{HE.Enc}(\text{pk}_{\text{HE}}, z_j; r''_{\text{HE},j}), \text{HE.Mul}(c_j, \text{ct}_{\text{HE},j})), \\ \text{ct}'_{\text{LE},j} &:= \text{LE.Subt}(\text{LE.Enc}(\text{pk}_{\text{LE}}, z_j; r''_{\text{LE},j}), \text{LE.Mul}(c_j, \text{ct}_{\text{LE},j})). \end{aligned}$$

The simulator outputs

$$(\text{com}, \{c_j\}_{j \in [\ell]}, (z_j, r''_{\text{crs},j}, r''_{\text{HE},j}, r''_{\text{LE},j})_{j \in [\ell]}),$$

where com contains the ciphertexts just defined and the simulated bit proofs. All range checks hold, and all three ciphertext equations hold by construction.

We compare this distribution with an honest transcript using the following hybrids. Throughout, the statement, its witness, and the challenge vector are fixed.

Hybrid H_0 (real CRS and transcript). Generate the CRS, commitment, and response honestly. Thus

$$z_j = u_j + c_j m_j, \quad r''_{X,j} = r'_{X,j} + c_j r_{X,j} \quad (X \in \{\text{crs}, \text{HE}, \text{LE}\}).$$

Hybrid H_1 (simulate the bit-proof CRS and proofs). Replace the honestly generated crs_{bit} by the output of the bit-proof simulator’s CRS generation, and replace every honest $\pi_{\text{bit},j}$ by a simulated proof for the same true statement $(\text{pk}_{\text{HE}}, \text{ct}_{\text{HE},j})$. Computational zero knowledge of NIZK_{bit} —applied to the joint CRS-and-proof distribution and hybridized over the ℓ coordinates—gives $H_0 \stackrel{\mathcal{C}}{\approx} H_1$.

Hybrid H_2 (flood the randomness responses). First observe that H_1 can be generated equivalently by computing each first-round mask ciphertext backwards from the honest response:

$$\begin{aligned} \text{ct}'_{\text{crs},j} &= \text{Subt}(\text{Enc}(\text{pk}_{\text{crs}}, z_j; r''_{\text{crs},j}), \text{Mul}(c_j, \text{ct}_{\text{crs},j})), \\ \text{ct}'_{\text{HE},j} &= \text{HE.Subt}(\text{HE.Enc}(\text{pk}_{\text{HE}}, z_j; r''_{\text{HE},j}), \text{HE.Mul}(c_j, \text{ct}_{\text{HE},j})), \\ \text{ct}'_{\text{LE},j} &= \text{LE.Subt}(\text{LE.Enc}(\text{pk}_{\text{LE}}, z_j; r''_{\text{LE},j}), \text{LE.Mul}(c_j, \text{ct}_{\text{LE},j})). \end{aligned}$$

Indeed, substituting $z_j = u_j + c_j m_j$ and $r''_{X,j} = r'_{X,j} + c_j r_{X,j}$ and using bi-homomorphism gives exactly $\text{ct}'_{X,j} = \text{Enc}_X(\text{pk}_X, u_j; r'_{X,j})$.

We now replace each response $r''_{X,j} = r'_{X,j} + c_j r_{X,j}$ by an independent uniform value in $\mathcal{R}_{\text{mask}}$, retaining the backwards computation above. For fixed c_j and $r_{X,j}$, the former distribution is a uniform interval of length R shifted by

$$0 \leq c_j r_{X,j} < CB.$$

By Lemma A.4, its statistical distance from a uniform value in $\mathcal{R}_{\text{mask}}$ is at most CB/R . Applying data processing to the backwards ciphertext computation and summing over three schemes and ℓ coordinates gives

$$\Delta(H_1, H_2) \leq \frac{3\ell CB}{R}.$$

Hybrid H_3 (flood the plaintext responses). Replace every $z_j = u_j + c_j m_j$ by an independent uniform $z_j \leftarrow \mathcal{U}$, again retaining the backwards computation of the mask ciphertexts. Since $m_j \in \{0, 1\}$,

$$0 \leq c_j m_j < C.$$

Hence Lemma A.4 and data processing give

$$\Delta(H_2, H_3) \leq \frac{\ell C}{U}.$$

Hybrid H_4 (remove the message from the extractable ciphertext). For each coordinate, replace $\text{ct}_{\text{crs},j} = \text{Enc}(\text{pk}_{\text{crs}}, m_j; r_{\text{crs},j})$ by a fresh encryption $\text{Enc}(\text{pk}_{\text{crs}}, 0)$. The associated $\text{ct}'_{\text{crs},j}$ is still computed backwards from this ciphertext. A standard hybrid over the ℓ coordinates reduces distinguishability of H_3 and H_4 to IND-CPA security of the CRS encryption: given the IND-CPA challenge ciphertext, the reduction performs the same public backwards computation to produce $\text{ct}'_{\text{crs},j}$. Thus $H_3 \stackrel{\mathcal{E}}{\approx} H_4$.

Hybrid H_4 is exactly the simulator's distribution. The two statistical losses are negligible by Equation (2), and the other two transitions are computationally indistinguishable. Therefore the real and simulated transcripts are computationally indistinguishable.

Under Fiat–Shamir, the simulator additionally programs H_{fs} at the resulting statement–commitment pair to return the chosen challenge vector. This does not change the straight-line transcript simulation described above. $\square$

Lemma A.4 (Adapted from [1, Lemma 2.1]). *Let $B_1 = B_1(\lambda)$, and $B_2 = B_2(\lambda)$ be positive integers and let $c \in [0, B_1]$ be a fixed integer. Let $z \leftarrow [0, B_2]$ be chosen uniformly at random. Then the distribution of z is statistically indistinguishable from that of $z + c$ as long as $B_1/B_2 = \nu(\lambda)$.*

Proof. To prove this lemma, for fixed c , we can bound the following:

$$\begin{aligned}
& \frac{1}{2} \sum_{x=0}^{B_2+c} |\Pr[z=x] - \Pr[z=x-c]| \\
&= \frac{1}{2} \left(\sum_{x=0}^{c-1} \underbrace{|\Pr[z=x] - \Pr[z=x-c]|}_{=\frac{1}{B_2+1}} \right) \\
&+ \sum_{x=c}^{B_2} \underbrace{|\Pr[z=x] - \Pr[z=x-c]|}_{=0} \\
&+ \sum_{x=B_2+1}^{B_2+c} \left(\underbrace{|\Pr[z=x] - \Pr[z=x-c]|}_{=0} \right) \\
&= \frac{1}{2} \left(\frac{c}{B_2+1} + \frac{c}{B_2+1} \right) = \frac{c}{B_2+1} \leq \frac{B_1}{B_2}
\end{aligned}$$

For $B_1/B_2 = \nu(\lambda)$ this yields the lemma. □