

# Better Security Proofs for X3DH and XHMQV

Jiawei Bao, Jiaxin Pan , and Runzhi Zeng 

University of Kassel, Kassel, Germany

`{jiawei.bao, jiaxin.pan, runzhi.zeng}@uni-kassel.de`

**Abstract.** The Signal protocol is used by billions of users daily and recognized as the gold standard for end-to-end encrypted messaging. Its initial handshake protocol X3DH uses XEdDSA to sign its semi-static key and allows parties to derive a session key asynchronously. The protocol is implemented over Curve25519, relying on the assumed 128-bit hardness for solving Discrete Logarithms (DL). Previous non-tight reductions incur a large loss in the number of sessions, and the resulting concrete security guarantees fall far below the intended 128-bit security level. This motivates the development of tight security bounds for these protocols.

In this paper, we improve the security analysis of X3DH and its recent enhancement XHMQV (Fiedler et al., CRYPTO’25) by providing tight security reductions under multi-user Diffie–Hellman (DH) assumptions (Kiltz et al., CT-RSA’23) in the Random Oracle Model. Unlike prior work, our proofs are in the more realistic multi-Test setting. The variant of X3DH that we analyze hashes additional context into the session key. Although this modification is minor, it yields tight security bounds and provides a stronger justification for the use of Curve25519. In light of our results, the Signal developers plan to adopt the same modification.

**Keywords.** Secure messaging, initial handshake protocols, signature, tight security

## 1 Introduction

In the paradigm of provable security, a cryptographic protocol  $\Pi$  is proven secure via a reduction that turns an adversary  $\mathcal{A}$  against breaking  $\Pi$  in a security model  $\mathsf{M}$  to another adversary  $\mathcal{R}$  breaking a hardness assumption  $\mathsf{P}$ . Assuming that the run time of  $\mathcal{A}$  and  $\mathcal{R}$  are roughly the same, we establish the following relation between the success probability of  $\mathcal{A}$  and  $\mathcal{R}$  as

$$\epsilon_{\mathcal{A}} \leq \ell \cdot \epsilon_{\mathcal{R}}. \quad (1)$$

where  $\epsilon_{\mathcal{A}}$  and  $\epsilon_{\mathcal{R}}$  are the winning advantages of  $\mathcal{A}$  and  $\mathcal{R}$ , and  $\ell$  is called the security loss. Asymptotically, if  $\ell$  is a polynomial, then we can conclude that  $\Pi$  is secure, since  $\epsilon_{\mathcal{R}}$  is assumed to be negligible. For concrete security, we call the reduction  $\mathcal{R}$  tight if  $\ell$  is a small constant, or non-tight if  $\ell$  is a polynomial dependent on the adversary’s oracle queries.

To provide better security guarantees for a protocol, we aim to work in a realistic security model and derive a security proof that provides a reasonable

level of concrete security. In this paper, we want to provide security proofs for the extended triple Diffie–Hellman (X3DH) [45] protocol and its variant XHMQV [24] under more realistic security models with tight bounds.

### 1.1 Motivation for New Security Proofs for X3DH

The X3DH protocol [45] is the initial handshake protocol running in the Signal protocol [59]. The Signal protocol is the “gold standard” for end-to-end encrypted messaging, and it is used by major secure messaging apps such as Facebook Messenger, Google Messages, Signal, and WhatsApp. It has been used by billions of people daily. Its security heavily relies on the session key that is initiated by the X3DH protocol. Hence, the security of X3DH is highly critical for today’s digital society.

In the X3DH protocol, two users communicate in an asynchronous way, namely, all users can upload a “pre-key bundle” to the Signal server, and an initiator of a conversation can fetch this pre-key bundle, execute its part of the X3DH protocol and start sending messages, while its receiver can later complete its part of the handshake to decrypt the messages. In other words, the initiator and receiver do not need to be online simultaneously. We sketch the protocol execution in Figure 1.

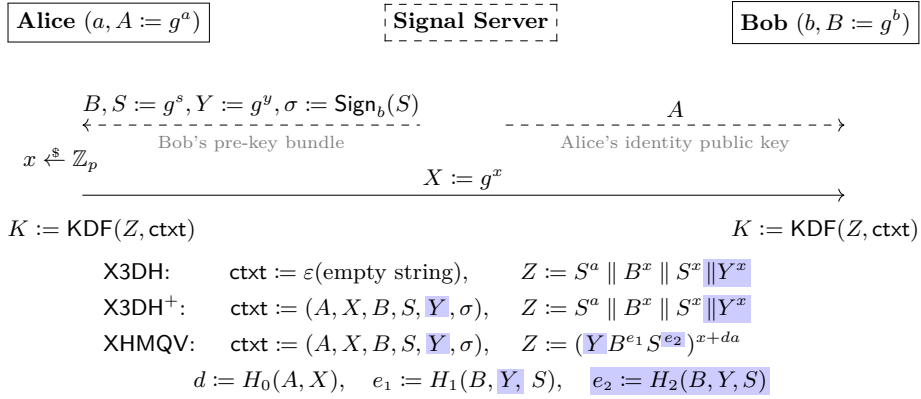


**Fig. 1.** Overview of a (simplified) execution of X3DH, X3DH<sup>+</sup>, or XHMQV, involving long-term keys  $a$  and  $b$ , semi-static keys  $s$ , ephemeral keys  $x$ , and one-time pre-keys  $y$ , as well as hash functions  $h$ . In the reduced mode, highlighted elements will be ignored.

The pre-key bundle contains the long-term key  $g^b$ , semi-static key  $g^s$ , and a signature on  $g^s$  under the secret key  $b$ . The security goal of X3DH is often referred to as “maximum-exposure” security, namely, the agreed session key remains pseudo-random, even if an adversary can corrupt some (but not all) of users’ long-term keys, semi-static keys, and ephemeral randomness. This is achieved by deriving the session key from the concatenation of 4 Diffie–Hellman

(DH) secrets  $g^{as}$ ,  $g^{xb}$ ,  $g^{xs}$ , and  $g^{xy}$ . Each of them combines two long-term ( $g^a$ ,  $g^b$  for users Alice and Bob), medium-term ( $g^s$  for user Bob only), or ephemeral DH shares ( $g^x$ ,  $g^y$ ). The value  $g^{xy}$  is optional and not included in the reduced mode.

The security of X3DH has been formally proven in [16,17], and related designs have been extensively studied in [40,60,31,32,21,13,25,11,19,23,33,24]. However, we identify several insufficiencies in the current analysis that lead to limited concrete security guarantees.

**THE PROBLEM OF LARGE SECURITY LOSS.** One of the major issues of X3DH is that it has a huge security loss. In Table 1, we detail the previous bounds and ours. For instance, the security proof in [16,17] has a loss of  $O(N_{\text{LT}}^2 N_{\text{ses}}^2)$  where  $N_{\text{LT}}$  and  $N_{\text{ses}}$  are the numbers of long-term keys and sessions. It is realistic to estimate that  $N_{\text{LT}} \approx 2^{30}$  and  $N_{\text{ses}} \approx 2^{32}$ .<sup>1</sup> For an instantiation of X3DH using a 128-bit security curve, as deployed in Signal, this proof yields only about 4 bits of concrete security which is not a meaningful level of security guarantees.

| Protocol                         | Security Bound                                                                                               | Remark                       | GGM Bound w/ Multi-TEST<br>$\epsilon(t_{\text{ON}}, t_{\text{OFF}})$ | Bit security |
|----------------------------------|--------------------------------------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------------|--------------|
| X3DH [16]                        | $O(N_{\text{LT}}^2 N_{\text{ses}}^2) \cdot \epsilon_{\text{GDH}}$                                            | Single-TEST,<br>no signature | $\frac{t_{\text{ON}}^5 t_{\text{OFF}}^2}{p}$                         | 0            |
| X3DH [23]                        | $O(N_{\text{ses}}^2) \cdot \epsilon_{\text{GDH}} + O(N_{\text{LT}}) \cdot \epsilon_{\text{sig}}$             | Single-TEST,<br>no key reuse | $\frac{t_{\text{ON}}^3 t_{\text{OFF}}^2}{p}$                         | 32           |
| XHMQV [24]                       | $O(N_{\text{ses}}^2) \cdot \epsilon_{\text{crGDH}} + O(N_{\text{LT}}) \cdot \epsilon_{\text{EUF-opCMA-DDH}}$ | Single-TEST,<br>key reuse    | $\frac{t_{\text{ON}}^3 t_{\text{OFF}}^2}{p}$                         | 32           |
| X3DH <sup>+</sup><br>(Theorem 1) | $\epsilon_{\text{EUF}^c\text{-ICMA-DDH}} + \epsilon_{\text{CorrGapCDH}}$                                     | Multi-TEST,<br>key reuse     | $\frac{t_{\text{OFF}}^2}{p}$                                         | 128          |
| XHMQV<br>(Theorem 2)             | $\epsilon_{\text{EUF}^c\text{-ICMA-DDH}} + \epsilon_{\text{CorrCRGapCDH}}$                                   | Multi-TEST,<br>key reuse     | $\frac{t_{\text{OFF}}^2 + t_{\text{ON}}^2 t_{\text{OFF}}}{p}$        | 128          |

**Table 1.** Overview on the security of X3DH, X3DH<sup>+</sup>, and XHMQV.  $N_{\text{LT}}$  and  $N_{\text{ses}}$  are the numbers of parties and the sessions.  $\epsilon_{\text{sig}}$  is the security bound of the signature scheme without considering key reuse,  $\epsilon_{\text{GDH}}$  and  $\epsilon_{\text{crGDH}}$  are the security bounds of the GapCDH and CRGapCDH assumptions in the single-user, single-challenge setting. “**Bit security**” is estimated by the respective bounds over generic groups of order  $p \approx 2^{256}$  (assuming the underlying curve has 128-bit hardness), and we consider a setting with  $t_{\text{ON}} \approx 2^{32}$  and  $t_{\text{OFF}} \lesssim 2^{128}$ .

However, this proof does not count the underlying signature security, and it considers adversaries in a single-TEST model, meaning that an adversary can only target at most one session. In the real world, an adversary is confronted with many sessions and should be able to test multiple sessions to increase its success probability. Hence, a security model with multiple TEST or challenge queries is more realistic and widely accepted in the context of public-key encryption schemes [3,28,34] and authenticated key exchange protocols [18,36,51,50].

<sup>1</sup> Signal has about 70–100 million monthly active users, cf. <https://backlinko.com/signal-stats>.

Introducing multiple TEST queries does not seem technically hard. In fact, these two notions are asymptotically equivalent, and the single-TEST security implies the multi-TEST security via a non-tight reduction. This non-tight reduction will introduce an additional multiplicative loss factor  $N_{\text{TEST}}$ , the number of TEST queries.  $N_{\text{TEST}} \approx N_{\text{ses}}$  because an adversary can ask TEST as many times as possible, and a real-world one may target almost all sessions. In the aforementioned example, even if one implements X3DH with Curve448 (providing 224-bit DL-hardness), it can give 68-bit security ( $N_{\text{TEST}} N_{\text{LT}}^2 N_{\text{ses}}^2 \cdot \epsilon_{\text{GDH}} \approx 2^{156} \times 2^{-224} \approx 2^{-68}$ ), which is still not a meaningful concrete security.

We observe that one can derive a tighter bound for X3DH, as  $O(N_{\text{TEST}} N_{\text{ses}}^2)$  in the multi-TEST model, by following the classical part of the analysis of PQXDH in [23]. But even with this tighter bound it only gives 32-bit security (assuming 128-bit DL-hardness) and 128-bit security (assuming 224-bit DL-hardness). Hence, it is practically important to provide a *tight* security proof for X3DH, a protocol that is used by millions of users.

**INCOMPLETE SECURITY ANALYSIS OF XEDDSA UNDER KEY REUSE.** As mentioned earlier, the previous analyses of X3DH [16,17] do not take the security of XEdDSA under key reuse into account. More precisely, X3DH uses the XEdDSA scheme [54] to sign the semi-static key in the pre-key bundle, and the long-term signing key  $b$  is *reused* in the X25519 key exchange protocol to derive the session key  $K$ . In this setting, the standard existential unforgeability (EUF-CMA) is not sufficient for proving the security of X3DH because during the proof of X3DH we need to do a reduction to break EUF-CMA. Due to the key reuse issue (namely, the signing and key exchange long-term keys are the same), this reduction does not know the signing key  $b$  to simulate the corresponding session keys.

Very recently, the security requirement to address this issue has been formally identified by Fiedler et al. [24], although the key reuse has been extensively studied in the public-key encryption and signature settings [30,2,20,52,53]. The work of Fiedler et al. formalized this “key reuse” requirement as a more fine-grained security notion for digital signature schemes, namely, one-per message unforgeability with a DDH oracle (EUF-opCMA-DDH). However, Fiedler et al. did not analyze whether XEdDSA is EUF-opCMA-DDH secure, although the EUF-CMA security of XEdDSA is implied by the work of Brendel et al. [12].

In the EUF-opCMA-DDH security, a DDH oracle is provided to check if  $(X = g^x, Y = g^y, Z)$  is a DH tuple, namely, checking  $Z = g^{xy}$ . Similar to the standard EUF-CMA security, an adversary  $\mathcal{A}$  receives a single public key and can ask for signatures on the adaptive chosen messages. To break the security,  $\mathcal{A}$  needs to output a forgery. EUF-opCMA-DDH is slightly weaker than EUF-CMA in the sense that for each message  $\mathcal{A}$  can ask for signature at most once. Jumping ahead, we close the gap in [24] by showing that XEdDSA satisfies a (slightly) more realistic version of EUF-opCMA-DDH security with a tighter reduction than [12].

## 1.2 Our Contributions

In this paper, we improve the security proofs of X3DH and its recent enhancement XHMQV. XHMQV was proposed to strengthen the “maximum-exposure”

security of X3DH. For instance, XHMQV remains secure even if the initiator (Alice) ephemeral randomness  $x$  and the responder (Bob) semi-static secret  $s$  are compromised, while it is not the case for X3DH.

**TIGHT SECURITY FOR X3DH<sup>+</sup> AND XHMQV.** Our main contributions are *tight* security proofs for XHMQV and a variant of X3DH (called X3DH<sup>+</sup>) in the more realistic multi-TEST model under the multi-user Diffie-Hellman (DH) assumptions [39] and the multi-user EUF-opCMA-DDH security in the Random Oracle Model (ROM). Specifically, assuming the EUF<sup>c</sup>-1CMA-DDH security of XEdDSA (which will be discussed more), we prove in the ROM that:

- The Gap DH with Corruption (CorrGapCDH) assumption [39] *tightly* implies the security of X3DH<sup>+</sup>; and
- The Challenge-Response Gap DH with Corruption (CorrCRGapCDH) assumption [39] *tightly* implies the security of XHMQV.

Unlike the previous proofs on X3DH in [16,17], our proof addresses the issue of key reuse. The original proof for XHMQV in [24] is in the single-TEST model, and extending it to the multi-TEST model naturally requires an additional linear loss of  $N_{\text{TEST}}$ . Actually, they used the single-user, single-challenge version of CorrCRGapCDH (aka CRGapCDH), and already in the single-TEST model, their guessing strategy requires a security loss of  $O(N_{\text{ses}}^2)$ . Our proof to CorrCRGapCDH does not lose any of these factors. To be a bit more technical, our proof does not hide the factor  $N_{\text{TEST}}$  in the parameters of CorrCRGapCDH. One can see that  $N_{\text{TEST}}$  does not appear in Theorem 2, although it is an analysis in the multi-TEST model. An overview of our implications is given in Figure 2.

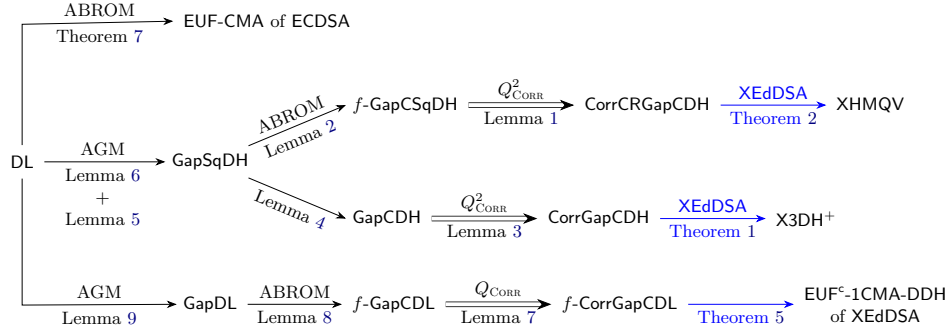


**Fig. 2.** Overview of our implications proven in this paper. Major contributions are highlighted with blue.  $Q_{\text{CORR}}$  is the number of corruption queries, which is smaller than the number of users. Single arrows represent tight implications, while double arrows indicate non-tight ones. Implications without specifying any model means in the “standard” model, meaning without using AGM or ABROM. Implications “ $f\text{-GapCSqDH} \rightarrow \text{CorrCRGapCDH}$ ” and “ $\text{GapCDH} \rightarrow \text{CorrGapCDH}$ ” do not even need the ROM.

PRACTICAL IMPACTS. The main novelty of our new proofs lies in their practical applicability. As a key enabler, we show the quantitative bound for the XEdDSA’s EUF<sup>c</sup>-1CMA-DDH security in the Generic Group Model (GGM) [58,46], which is optimal in the sense of matching the GGM-bound of the discrete logarithm (DL) problem. Given our tight implications, we can directly apply the concrete generic bounds of CorrGapCDH and CorrCRGapCDH as in [39] to derive those of X3DH<sup>+</sup> and XHMQV without compensating the security loss of the previous reductions. Our GGM-bounds of X3DH<sup>+</sup> and XHMQV are (almost) optimal, and thus they support the implementation of X3DH<sup>+</sup> and XHMQV without increasing the group sizes to compensate for the security loss of the previous reductions. Our GGM bounds and the corresponding concrete bit security are given in Table 1, and we will explain how they are derived.

The X3DH<sup>+</sup> protocol considered here is a (slight) variant of the currently deployed X3DH by hashing the “context” into the session key. We refer to Figure 1 for an overview and Section 3.2 for concrete details. The modification of hashing more information is rather minor in practice, compared to increasing the group size. This is confirmed by our private communication with the Signal development team. In light of our work on optimal GGM-bounds, they informed us that they will make this minor change to X3DH.

SECURITY OF XEDDSA WITH KEY REUSE. Our tight implications require the EUF-opCMA-DDH [24] security in a multi-user setting, namely, we additionally allow adversaries to receive multiple public keys and corrupt a subset of their corresponding signing keys. We denote this stronger notion as EUF<sup>c</sup>-1CMA-DDH.

Our proof for the EUF<sup>c</sup>-1CMA-DDH security of XEdDSA is modular: We first propose an algebraic assumption, Gap Circular Discrete Logarithms with Corruption ( $f$ -CorrGapCDL), and we show that  $f$ -CorrGapCDL tightly implies the EUF<sup>c</sup>-1CMA-DDH security of XEdDSA in the ROM. Then we analyze the concrete bound of  $f$ -CorrGapCDL in the GGM as  $\epsilon_{f\text{-CorrGapCDL}} \leq (Q_{\text{Op}} + N_{\text{LT}})^2/p$  (cf. Theorem 6), ignoring the constants and taking only the dominating term, where  $N_{\text{LT}}$  is the number of long-term signing keys. Recall that the GGM-bound of breaking DL is  $\epsilon_{\text{DL}} \approx Q_{\text{Op}}^2/p$ . We have  $\epsilon_{f\text{-CorrGapCDL}} \approx \epsilon_{\text{DL}}$ , since  $Q_{\text{Op}} \gg N_{\text{LT}}$ . Hence, our GGM-bound for the EUF<sup>c</sup>-1CMA-DDH security of XEdDSA is optimal and matches that of DL.

To provide more confidence in the new assumption  $f$ -CorrGapCDL, we show that it is implied by the DL assumption in the Algebraic Group Model (AGM) [27] and Algebraic Bijective Random Oracle Model (ABROM) [56] and Random Oracle Model (ROM) [6].  $f$ -CorrGapCDL is motivated by the Circular DL assumption of Cho et al. [15]. This part of our proof adapts the Cho et al. approach to remove the rewinding and the Loss approach [44, Section 5.3.2] to remove the use of the Gap oracle.

As an extension of our method in the AGM, we prove that the ECDSA signature scheme is tightly EUF-CMA secure in the ABROM. Although it is not directly related to X3DH, we think this tight proof is of independent interest, since ECDSA is one of the most widely deployed digital signature schemes. It is

standardized by the NIST and extensively used in TLS, cryptocurrencies such as Bitcoin and Ethereum, and secure hardware implementations.

**NEW HARDNESS ANALYSIS OF MULTI-USER GAP DH ASSUMPTIONS.** The original hardness analysis in [39] provides two analyses of these assumptions. The first one is giving direct bounds in the Generic Group Model (GGM). The second one is an implication by the Gap Computational DH (GapCDH) assumption [49], where an adversary is asked to solve CDH, given access to DDH. This implication for CorrCRGapCDH is done by using the rewinding techniques [55,4] and leads to a square-root loss.

To provide more confidence in these assumptions, we analyze them in the AGM and ABROM: We adapt the Circular-DL framework of Cho et al. to remove the use of rewinding for CorrCRGapCDH, and furthermore we remove the use of the Gap oracle in both CorrCRGapCDH and CorrGapCDH and prove that they are implied by DL in the AGM.

For the “ $f$ -GapCSqDH  $\rightarrow$  CorrCRGapCDH” implication, we extend the “Count-Corruptions-Not-Users” technique [5] to improve the loss from depending on the number of users to the number of corruptions. Similarly, it works for “GapCDH  $\rightarrow$  CorrGapCDH” and “ $f$ -GapCDL  $\rightarrow$   $f$ -CorrGapCDL” as well.

**GENERIC BOUNDS.** We follow the approach in [39] to show how our results can give optimal bounds for X3DH<sup>+</sup> and XHMQV and how our GGM bounds in Table 1 are derived. Firstly, the running time of an adversary  $\mathcal{A}$  in the GGM depends on the numbers of long-term key queries  $N_{\text{LT}}$ , sessions  $N_{\text{ses}}$ , TEST queries  $N_{\text{TEST}}$ , hash queries  $Q_{\text{Ro}}$ , and group operations  $Q_{\text{Op}}$ .  $\mathcal{A}$ ’s time  $t = t_{\text{ON}} + t_{\text{OFF}}$  can be divided into (1) the time  $t_{\text{ON}} = N_{\text{LT}} + N_{\text{ses}} + N_{\text{TEST}}$  that is spent on the “online” queries and (2) the time  $t_{\text{OFF}} = Q_{\text{Op}} + Q_{\text{Ro}}$  that is spent on the “offline” queries. The offline queries model the operations  $\mathcal{A}$  can do locally, offline. It is often the case  $t_{\text{OFF}}$  is much larger than  $t_{\text{ON}}$ . The GGM bounds in Table 1 are expressed as functions of  $t_{\text{ON}}$  and  $t_{\text{OFF}}$ , and for the previous bounds on X3DH [16,23], we take  $\epsilon_{\text{GDH}} \approx \epsilon_{\text{DL}} \approx t_{\text{OFF}}^2/p$  and  $\epsilon_{\text{sig}} \approx \epsilon_{\text{EUF-opCMA-DDH}}$  is roughly the same as the GGM bound for the XEdDSA’s EUF<sup>c</sup>-1CMA-DDH security,  $\epsilon_{\text{EUF}^c\text{-1CMA-DDH}} \leq (Q_{\text{Op}} + N_{\text{LT}})^2/p \approx t_{\text{OFF}}^2/p$  (assuming  $Q_{\text{Op}} \gg N_{\text{LT}}$ ), even though this is not proven in [16,23].

Secondly, in [39] it is shown that  $\epsilon_{\text{CorrGapCDH}} \approx t_{\text{OFF}}^2/p$  and  $\epsilon_{\text{CorrCRGapCDH}} \approx (t_{\text{OFF}}^2 + t_{\text{ON}}^2 t_{\text{OFF}})/p$ . Thirdly, according to [14], a scheme offers  $k$ -bit security if  $\epsilon/t \leq 2^{-k}$  for adversaries running in time  $t$  with  $1 \leq t = t_{\text{ON}} + t_{\text{OFF}} \leq 2^k$ . For our comparison in the GGM, we take  $\log_2(p) \approx 256$  (which is the case for Curve25519) and  $t_{\text{ON}} \approx 2^{32}$ . As long as  $t_{\text{ON}} + t_{\text{OFF}} \leq 2^{128}$ , our new bounds offer about 128-bit security for X3DH<sup>+</sup> and XHMQV, but not for the previous bounds. The previous bounds for X3DH offer about  $128 - 5 \log_2(t_{\text{ON}}) \approx -32$ -bit (meaning no concrete security guarantee) and  $128 - 3 \log_2(t_{\text{ON}}) = 32$ -bit security.

### 1.3 Related Work

**MULTI-USER GAP DH ASSUMPTIONS.** The CorrGapCDH and CorrCRGapCDH assumptions are proposed by Kiltz et al. [39] to show the tight security for NAXOS

[43], a simplified version of X3DH, and HMQV [41]. However, these protocols are classical authenticated key exchange protocols that assume synchronous interaction between parties. As a result, the security analysis in [39] does not directly apply to the secure messaging setting, where asynchronous communication must be supported. Moreover, the aforementioned classical AKE protocols do not have the issue of key reuse (as in X3DH and XHMQV).

**EXISTING SECURITY ANALYSIS FOR XEdDSA.** XEdDSA can be viewed as an instantiation of the Ed25519 signature scheme [10] (or, more generally, EdDSA [38]) that is compatible with the X25519 key exchange protocol. The EUF-CMA security of Ed25519 was proven by Brendel et al. [12]. We give the first proof that XEdDSA satisfies the  $\text{EUF}^c\text{-1CMA-DDH}$  security, which is needed for X3DH.

**EXTENSION TO RELATED WORK.** Very recently, Benedikt et al. [9] designed a new initial handshake protocol to improve the efficiency of X3DH, and it has similar efficiency as XHMQV. Its security proof is in the multi-TEST model and based on the Square DH assumption, assuming the AGM. However, its security bound contains a larger security loss than ours (which is roughly  $O(N_{\text{TEST}} q_{\text{KDF}}^2 N_{\text{ses}}^2)$ ). Using the Gap Oracle, the factor of  $N_{\text{TEST}} \cdot q_{\text{KDF}}^2$  may be removed. We are optimistic that we may propose a new algebraic assumption that can tightly imply their protocol and derive better GGM bounds.

Finally, we are optimistic that our techniques can be useful to tighten the classic security bound of PQXDH (as Theorem 5.1 in [23]) because PQXDH layers a post-quantum KEM on top of X3DH. Similar to  $\text{X3DH}^+$ , we also need to include the context into PQXDH. We give some brief ideas in Section 3.3 and leave the formal (and potentially tighter) security proof as an interesting future direction.

## 2 Preliminaries

**NOTATIONS.** Let  $[n]$  be the set of all the positive integers up to  $n$ . For a set  $S$ , we use  $r \xleftarrow{\$} S$  to denote that  $r$  is sampled uniformly at random from  $S$ , and use  $\mathcal{P}_t(S)$  to denote the family of all  $t$ -sized subsets of  $S$ . For a probabilistic algorithm  $\mathcal{A}$ , we write  $x \xleftarrow{\$} \mathcal{A}$  to denote its output. For a function  $f : \mathbb{A} \rightarrow \mathbb{B}$ , we use  $|f^{-1}(\beta)|$  to denote the size of the preimage of  $\beta$ , and define  $\text{MAX}_f := \max_{\beta \in \mathbb{B}} |f^{-1}(\beta)|$  as the largest size of the preimage set. Let  $\mathbf{T}(\mathcal{A})$  denote the running time of the algorithm  $\mathcal{A}$ .

**SECURITY GAMES.** We will use the code-based approach to describe our proofs and definitions, following [8, 39, 24]. In our games, all Boolean values are initialized to **false**, numerical values to 0, sets to  $\emptyset$ , strings to undefined  $\perp$ . We write an empty string as  $\varepsilon$ . A procedure terminates once it has returned an output.

**IDEALIZED MODELS.** Our security proofs are considered in the Random Oracle Model (ROM), where a hash function is modeled as a random function and can be accessed by an adversary  $\mathcal{A}$  via an oracle  $H$  that (consistently) outputs random elements in the range of the hash.

For the rest of this paper, we fix parameters  $\text{par} := (\mathbb{G}, g, p)$ , where  $\mathbb{G} := \langle g \rangle$  is a cyclic group with prime order  $p$ . We consider the following idealized models of  $\mathbb{G}$  from the previous work.

**Definition 1 (Algebraic Group Model [27]).** *In the Algebraic Group Model (AGM), for every group element  $Z \in \mathbb{G}$  output by an algebraic adversary  $\mathcal{A}$ , the corresponding algebraic representation must be provided. This  $Z$  can be a part of an oracle query from  $\mathcal{A}$ . Specifically, let  $X_1, \dots, X_m \in \mathbb{G}$  be all the group elements that  $\mathcal{A}$  has received so far. Besides the group element  $Z$ ,  $\mathcal{A}$  must also provide a representation vector  $\vec{a} := (a_1, \dots, a_m) \in \mathbb{Z}_p^m$  such that  $Z = \prod_{i=1}^m X_i^{a_i}$ .*

**Definition 2 (Semi-Injective Function [15]).** *Let  $\mathbb{A}$  be a finite set. A function  $\varphi : \mathbb{G} \rightarrow \mathbb{A}$  is semi-injective if its range  $\varphi(\mathbb{G}) \subseteq \mathbb{A}$  is efficiently decidable, and  $\varphi$  is either injective or 2-to-1 with  $\varphi(X) = \varphi(Y)$  implying  $X = Y$  or  $X = Y^{-1}$ .*

**Definition 3 (Algebraic Bijective Random Oracle Model [56]).** *In the Algebraic Bijective Random Oracle Model (ABROM), an adversary can only evaluate the conversion function  $f : \mathbb{G} \rightarrow \mathbb{Z}_p^2$  by querying the bijective random oracle BRO with the algebraic representations. Specifically, it decomposes  $f$  into three independent functions as  $f = \Psi \circ \Pi \circ \varphi$ , where  $\varphi : \mathbb{G} \rightarrow \{0, 1\}^L$  is a semi-injective function,  $\Pi : \{0, 1\}^L \rightarrow [2^L]$  is modeled as a bijective random oracle, and  $\Psi : [2^L] \rightarrow \mathbb{Z}_p$  is an arbitrary function.*

DISCUSSION ABOUT GGM, ABROM AND AGM. [56] points out that the ABROM aims to minimize the use of AGM and has been proposed as a reasonable model for analyzing the security of ECDSA. In the ABROM, the adversary is only required to provide algebraic representations for the group elements that are queried to the oracle BRO, whereas in the AGM the adversary must provide algebraic representations for all output group elements, including those sent as queries to the oracles. Therefore, the ABROM is a strictly weaker model than the AGM. Regarding the relation between the AGM and the generic group model (GGM), [62] shows that the lifting lemma in [27] fails under Shoup's GGM framework [58]. Later, [61] compares Shoup's GGM and Maurer's GGM [46], and shows that these two models are not equivalent. On the other hand, [35] proposes a lifting lemma for reductions constructed within Maurer's GGM.

## 2.1 Definitions of Assumptions

We recall some basic Diffie-Hellman (DH) Assumptions and the Multi-User Diffie-Hellman (DH) Assumptions proposed by Kiltz et al. [39], and we will use them later to prove security of X3DH<sup>+</sup> and XHMQV protocols.

**Definition 4 (Basic Assumptions).** *We define the attack games for the Gap Computational Diffie-Hellman (GapCDH) [49, 1], the Gap Square Diffie-Hellman (GapSqDH), the Square Diffie-Hellman (SqDH), the Gap Discrete Logarithm (GapDL) (which was implicitly defined in Section 2.2 of [24]), and the*

<sup>2</sup> The original preimage domain of  $f$  is  $\mathbb{G}^*$ , here without loss of generality, we use the modified model in [15] where the input domain is  $\mathbb{G}$ .



| $f\text{-GapCDL}_{Q_{\text{DDH}}}^{\mathcal{A}}, f\text{-GapCSqDH}_{Q_{\text{DDH}}}^{\mathcal{A}}$ | $f\text{-CorrGapCDL}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\mathcal{A}}$                                                                                                   |
|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01 $x \xleftarrow{\$} \mathbb{Z}_p$                                                                | 08 <b>for</b> $i \in [n]$                                                                                                                                                  |
| 02 $X := g^x$                                                                                      | 09 $x_i \xleftarrow{\$} \mathbb{Z}_p; X_i := g^{x_i}$                                                                                                                      |
| 03 $(R, z) \xleftarrow{\$} \mathcal{A}^{\text{DDH}}(X)$ // $f\text{-GapCDL}$                       | 10 $(R, z) \xleftarrow{\$} \mathcal{A}^{\text{CORR, DDH}}(X_1, \dots, X_n)$                                                                                                |
| 04 $Z := g^z$ // $f\text{-GapCDL}$                                                                 | 11 $Z := g^z$                                                                                                                                                              |
| 05 $(R, Z) \xleftarrow{\$} \mathcal{A}^{\text{DDH}}(X)$ // $f\text{-GapCSqDH}$                     | 12 <b>return</b> $\left[ \begin{array}{c} \exists i^* \in ([n] \setminus \mathcal{L}_X) \\ \text{s.t. } f(R) \neq 0 \wedge Z = R \cdot X_{i^*}^{f(R)} \end{array} \right]$ |
| 06 <b>return</b> $\llbracket f(R) \neq 0 \wedge Z = R \cdot X^{f(R)} \rrbracket$                   |                                                                                                                                                                            |
| <b>Oracle</b> $\text{DDH}(X_\ell, Y_\ell, Z_\ell)$                                                 | <b>Oracle</b> $\text{CORR}(X_i) \text{ // } i \in [n]$                                                                                                                     |
| 07 <b>return</b> $\llbracket Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)} \rrbracket$                   | 13 $\mathcal{L}_X := \mathcal{L}_X \cup \{i\}$                                                                                                                             |
|                                                                                                    | 14 <b>return</b> $x_i$                                                                                                                                                     |

**Fig. 4.** Games  $f\text{-GapCDL}$ ,  $f\text{-CorrGapCDL}$ , and  $f\text{-GapCSqDH}$  over the group  $\mathbb{G}$  in Definition 5.

Here we require  $|f^{-1}(0)|$  to be small as in [15], otherwise a trivial attack exists when  $f(R) = 0$ . For an elliptic-curve group  $\mathbb{G}$ , the assumptions hold for the ECDSA conversion function  $f$  [48], which maps a point on the curve to its  $x$ -coordinate.

We follow the definitions of the multi-user CDH assumptions given in [39] and present them in Definition 6. Moreover, we apply their lemma in Lemma 3.

**Definition 6 (Multi-User CDH assumptions [39]).** *The games  $\text{CorrGapCDH}$  and  $\text{CorrCRGapCDH}$  are given in Figure 5, where  $\text{DDH}$  is the Gap oracle,  $\text{CORR}$  is the corruption oracle, and  $\text{CH}$  is the Challenge oracle.  $Q_{\text{DDH}}$  and  $Q_{\text{CH}}$  denote the number of queries to the oracles  $\text{DDH}$  and  $\text{CH}$ , respectively. Let  $n$  be the number of input challenges, and the adversary  $\mathcal{A}$  makes  $Q_{\text{CORR}}$  queries to the corruption oracle  $\text{CORR}$ . We define the advantage functions as:*

$$\begin{aligned} \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{CorrGapCDH}}(\mathcal{A}) &= \Pr[\text{CorrGapCDH}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\mathcal{A}} \Rightarrow 1], \\ \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, Q_{\text{CH}}}^{\text{CorrCRGapCDH}}(\mathcal{A}) &= \Pr[\text{CorrCRGapCDH}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, Q_{\text{CH}}}^{\mathcal{A}} \Rightarrow 1]. \end{aligned}$$

| $\text{CorrGapCDH}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\mathcal{A}}, \text{CorrCRGapCDH}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, Q_{\text{CH}}}^{\mathcal{A}}$                                                                                                    | <b>Oracle</b> $\text{DDH}(X_\ell, Y_\ell, Z_\ell)$                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| 01 <b>for</b> $i \in [n]$                                                                                                                                                                                                                                        | 07 <b>return</b> $\llbracket Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)} \rrbracket$ |
| 02 $a_i \xleftarrow{\$} \mathbb{Z}_p; A_i := g^{a_i}$                                                                                                                                                                                                            | <b>Oracle</b> $\text{CORR}(A_i) \text{ // } i \in [n]$                           |
| 03 $Z \xleftarrow{\$} \mathcal{A}^{\text{DDH, CORR}}(A_1, \dots, A_n)$ // $\text{CorrGapCDH}$                                                                                                                                                                    | 08 $\mathcal{L}_A := \mathcal{L}_A \cup \{i\}$                                   |
| 04 $Z \xleftarrow{\$} \mathcal{A}^{\text{DDH, CORR, CH}}(A_1, \dots, A_n)$ // $\text{CorrCRGapCDH}$                                                                                                                                                              | 09 <b>return</b> $a_i$                                                           |
| 05 <b>return</b> $\left[ \begin{array}{c} \exists (i^*, j^*) \in (([n] \setminus \mathcal{L}_A)^2) \\ \text{s.t. } i^* \neq j^* \wedge Z = A_{i^*}^{a_{j^*}} \end{array} \right]$ // $\text{CorrGapCDH}$                                                         | <b>Oracle</b> $\text{CH}(R_k \in \mathbb{G})$                                    |
| 06 <b>return</b> $\left[ \begin{array}{c} \exists (i^*, j^*, k^*) \in (([n] \setminus \mathcal{L}_A)^2 \times [Q_{\text{CH}}]) \\ \text{s.t. } i^* \neq j^* \wedge Z = (A_{i^*}^{h_{k^*}} \cdot R_{k^*})^{a_{j^*}} \end{array} \right]$ // $\text{CorrCRGapCDH}$ | 10 <b>return</b> $h_k \xleftarrow{\$} \mathbb{Z}_p$                              |

**Fig. 5.** Games  $\text{CorrGapCDH}$  and  $\text{CorrCRGapCDH}$  over the group  $\mathbb{G}$  in Definition 6.

| EUFCMA $_{Q_{\text{sig}}}^A$                                                                                             | Oracle SIGN( $m$ )                                                   |
|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| 01 $(vk, sk) \leftarrow \text{KG}$                                                                                       | 04 $\sigma \leftarrow \text{Sign}(sk, m)$                            |
| 02 $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\text{SIGN}}(vk)$                                                            | 05 $\mathcal{S}_{\text{sig}} := \mathcal{S}_{\text{sig}} \cup \{m\}$ |
| 03 <b>return</b> $\llbracket m^* \notin \mathcal{S}_{\text{sig}} \wedge \text{Verify}(vk, \sigma^*, m^*) = 1 \rrbracket$ | 06 <b>return</b> $\sigma$                                            |

**Fig. 6.** The EUF-CMA game in Definition 7

## 2.2 Security Definitions for Signatures

We first recall the standard existential unforgeability against chosen-message attacks (EUF-CMA) definition for digital signature schemes in Definition 7.

**Definition 7 (EUF-CMA).** *Let  $\Sigma$  be a signature scheme and EUF-CMA be the security game defined in Figure 6. For any PPT adversary  $\mathcal{A}$  making  $Q_{\text{sig}}$  queries to SIGN, we define its EUF-CMA advantage in breaking  $\Sigma$  as*

$$\text{Adv}_{Q_{\text{sig}}}^{\text{EUF-CMA}}(\mathcal{A}) := \Pr \left[ \text{EUF-CMA}_{Q_{\text{sig}}}^A = 1 \right].$$

EUFC-1CMA-DDH. When proving the security of Signal’s X3DH handshake and XHMQV, a weaker form of EUF-CMA is sufficient, since each semi-static public key will only be signed once by a user. Hence, following [24], we consider the one-per message unforgeability notion, where an adversary is restricted to ask at most one signature per message. Moreover, to obtain a clean security reduction, we consider a variant that considers a multi-user setting and allows the adversary to corrupt some users and obtain their secret keys. Then, the forger  $\mathcal{A}$  wins if it can produce a valid signature under an uncorrupted user’s public key on a message that has not been signed by the uncorrupted user before. Finally, the definition in [24] also allows the adversary to query a DDH oracle of the underlying group to address the key reuse issue in X3DH and XHMQV. We present the formal definition of EUFC-1CMA-DDH in Figure 7.

**Definition 8 (EUFC-1CMA-DDH).** *Let  $\Sigma$  be a signature scheme based on a group  $\mathbb{G}$  (with order  $p$  and a public generator  $g$ ). Consider the security game EUFC-1CMA-DDH defined in Figure 7. For any PPT adversary  $\mathcal{A}$  making  $Q_{\text{sig}}$ ,  $Q_{\text{cor}}$ , and  $Q_{\text{ddh}}$  queries to SIGN, CORR, and DDH oracles, respectively, we define its EUFC-1CMA-DDH advantage in breaking  $\Sigma$  as*

$$\text{Adv}_{N, Q_{\text{sig}}, Q_{\text{cor}}, Q_{\text{ddh}}}^{\text{EUFC-1CMA-DDH}}(\mathcal{A}) := \Pr \left[ \text{EUFC-1CMA-DDH}_{N, Q_{\text{sig}}, Q_{\text{cor}}, Q_{\text{ddh}}}^A = 1 \right].$$

$\delta_{\text{sim}}$ -SIMULATABILITY. In Signal’s X3DH protocol, the long-term key is used both for key exchange and for signing pre-key bundles. Following [24], to stay as close as possible to the original protocol, we use the  $\delta_{\text{sim}}$ -simulatability definition (cf. Definition 9), which states that, when allowing to program the random oracle, there is an algorithm that can efficiently simulate signatures that are statistically close to real signatures.

| $\text{EUF}^c\text{-1CMA-DDH}_{N, Q_{\text{sig}}, Q_{\text{corr}}, Q_{\text{ddh}}}^A$                                                                                                                      | <b>Oracle</b> $\text{SIGN}(i, m)$                 | <b>Oracle</b> $\text{CORR}(i \in [N])$                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|------------------------------------------------------------------------|
| 01 For $i \in [N]$                                                                                                                                                                                         | 05 if $\mathcal{L}_{\text{sig}}[i, m] = \perp$    | 09 $\mathcal{S}_{\text{corr}} := \mathcal{S}_{\text{corr}} \cup \{i\}$ |
| 02 $(vk_i = g^{sk_i}, sk_i) \leftarrow \text{KG}$                                                                                                                                                          | 06 $\sigma \xleftarrow{\$} \text{Sign}(sk_i, m)$  | 10 <b>return</b> $sk_i$                                                |
| 03 $(i^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\text{SIGN, CORR, DDH}}((vk_i)_{i \in [N]})$                                                                                                              | 07 $\mathcal{L}_{\text{sig}}[i, m] := \sigma$     | <b>Oracle</b> $\text{DDH}(X, Y, Z)$                                    |
| 04 <b>return</b> $\left[ \begin{array}{l} i^* \notin \mathcal{S}_{\text{corr}} \wedge \mathcal{L}_{\text{sig}}[i^*, m^*] = \perp \\ \wedge \text{Verify}(vk_{i^*}, \sigma^*, m^*) = 1 \end{array} \right]$ | 08 <b>return</b> $\mathcal{L}_{\text{sig}}[i, m]$ | 11 <b>return</b> $[Z = X^{\text{DL}_g(Y)}]$                            |

**Fig. 7.** The  $\text{EUF}^c\text{-1CMA-DDH}$  game in Definition 8

| $\text{REAL}_{Q_{\text{sig}}}^A$                                                                     | $\text{IDEAL}_{Q_{\text{sig}}}^A$                                                                    | <b>Oracle</b> $\text{RO}(q)$ // In $\text{IDEAL}$                       |
|------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| 01 $(vk, sk) \leftarrow \text{KG}, \mathcal{L}_{\text{RO}} := \emptyset$                             | 07 $(vk, sk) \leftarrow \text{KG}, \mathcal{L}_{\text{RO}} := \emptyset$                             | 14 <b>if</b> $\exists h$ s.t. $(q, h) \in \mathcal{L}_{\text{RO}}$      |
| 02 $(m_1, \dots, m_{Q_{\text{sig}}}) \leftarrow \mathcal{A}$                                         | 08 $(m_1, \dots, m_{Q_{\text{sig}}}) \leftarrow \mathcal{A}()$                                       | 15 <b>return</b> $h$                                                    |
| 03 <b>for</b> $i \in \{1, \dots, Q_{\text{sig}}\}$                                                   | 09 <b>for</b> $i \in \{1, \dots, Q_{\text{sig}}\}$                                                   | 16 $h \xleftarrow{\$} \text{RO.image}$                                  |
| 04 $\sigma_i \xleftarrow{\$} \text{Sign}^{\text{RO}}(sk, m_i)$                                       | 10 $(\sigma_i, q_i, h_i) \xleftarrow{\$} \text{SIMsign}(vk, m_i)$                                    | 17 $\mathcal{L}_{\text{RO}} := \mathcal{L}_{\text{RO}} \cup \{(q, h)\}$ |
| 05 $\beta^* \leftarrow \mathcal{A}^{\text{RO}}(vk, sk, \{\sigma_i\}_{1 \leq i \leq Q_{\text{sig}}})$ | 11 $\mathcal{L}_{\text{RO}} := \mathcal{L}_{\text{RO}} \cup \{(q_i, h_i)\}$                          | 18 <b>return</b> $h$                                                    |
| 06 <b>return</b> $\beta^*$                                                                           | 12 $\beta^* \leftarrow \mathcal{A}^{\text{RO}}(vk, sk, \{\sigma_i\}_{1 \leq i \leq Q_{\text{sig}}})$ |                                                                         |
|                                                                                                      | 13 <b>return</b> $\beta^*$                                                                           |                                                                         |

**Fig. 8.** Security game for the  $\delta_{\text{sim}}$ -simulatability in Definition 9.

**Definition 9** ( $\delta_{\text{sim}}$ -Simulatability [24, Definition 3]). *Let  $\Sigma$  be a signature scheme whose signing algorithm queries a random oracle  $\text{RO}$ . Consider the two games defined in Figure 8. We say  $\Sigma$  is  $\delta_{\text{sim}}$ -simulatable if there exists an algorithm  $\text{SIMsign}$  such that for any (unbounded) stateful adversary  $\mathcal{A}$  and  $Q_{\text{sig}} \in \mathbb{N}$ , we have*

$$\left| \Pr \left[ \text{REAL}_{Q_{\text{sig}}}^A = 1 \right] - \Pr \left[ \text{IDEAL}_{Q_{\text{sig}}}^A = 1 \right] \right| \leq \delta_{\text{sim}}.$$

In [24], the authors show that ECDSA has  $\frac{Q_{\text{sig}}}{p}$ -simulatability and Schnorr signature scheme has perfect simulatability, where  $Q_{\text{sig}}$  is the number of signing queries and  $p$  is the order of the underlying group. For XEdDSA, we give the simulator in Figure 9. Different from ECDSA and Schnorr, XEdDSA uses a hash function  $G$  to derive the randomness  $r$  for signing, where  $G$  takes the secret key, the message, and a random string  $\text{INP}$  as input. We can see that the simulated signature  $(R, s)$  (Figure 9, right) has the same distribution as the original one (Figure 9, left), unless the adversary guesses the random  $\text{INP}$  correctly. Hence, the probability that  $\mathcal{A}$  sees such a difference is upper bounded by  $\frac{Q_G \cdot Q_{\text{sig}}}{2^{\text{len}}}$ , where  $Q_G$  is the number of  $\mathcal{A}$ 's  $\text{RO}$  queries to  $G$  and  $Q_{\text{sig}}$  is the number of signing queries, and XEdDSA is  $\delta_{\text{sim}}$ -simulatable with  $\delta_{\text{sim}} = \frac{Q_G \cdot Q_{\text{sig}}}{2^{\text{len}}}$ . In Signal's XEdDSA parameter,  $\text{len}$  is 512 (= 64 bytes).

In our security proof, for simplicity, we treat  $\delta_{\text{sim}}$  as some negligible term and ignore how the queries from the adversary affect it. For XEdDSA, even if we set  $Q_G := 2^{128}$  and  $Q_{\text{sig}} := 2^{128}$  (which is already extremely unlikely in practice), we still have  $\delta_{\text{sim}} = 2^{-256}$ , which is negligible compared to other terms (e.g., the DL advantage).

| <u>XEdDSA.Sign<sup>RO</sup>(sk = x, m)</u>            | <u>SIMsign(vk = X, m)</u>                                                            |
|-------------------------------------------------------|--------------------------------------------------------------------------------------|
| 01 $\text{INP} \xleftarrow{\$} \{0, 1\}^{\text{len}}$ | 07 $\text{INP} \xleftarrow{\$} \{0, 1\}^{\text{len}}$ // Useless for the simulation. |
| 02 $r := G(x \  m \  \text{INP})$                     | 08 $h \xleftarrow{\$} \mathbb{Z}_p$                                                  |
| 03 $R := g^r$                                         | 09 $s \xleftarrow{\$} \mathbb{Z}_p$                                                  |
| 04 $h := H(R \  X \  m)$                              | 10 $R := g^s \cdot (X)^{-h}$                                                         |
| 05 $s := r + h \cdot x$                               | 11 <b>return</b> ((R, s), (R, X, m), h) // Here q is (R, X, m).                      |
| 06 <b>return</b> $\sigma := (R, s)$                   |                                                                                      |

**Fig. 9.** Simulator for the XEdDSA signature scheme.

### 3 Bundled Authenticated Key Exchange

In this section, we first define the protocol syntax for the X3DH and XHMQV protocols. More specifically, we adopt the name *Bundled Authenticated Key Exchange* (BAKE) from [33] to capture the semantics and flow of Signal’s X3DH, where parties upload their pre-key bundles to a server that subsequently distributes them to peers, and define the syntax of BAKE based on the prior works [23, 24, 33, 50]. A main difference is that we use a session initialization algorithm `NewSession` to create sessions explicitly, which simplifies the security definition when handling the randomness.

Then, we briefly discuss the BAKE security model, which is essentially a multi-test version of the ones in [23, 24]. For readability, we present the complete security model in Appendix A.

Finally, we present the X3DH and XHMQV protocols in Section 3.2 and analyze their security in Theorems 1 and 2.

#### 3.1 Protocol Syntax and Security of BAKE

**DEFINING SESSIONS.** A session is denoted by  $\pi$ . If  $\pi$  is the  $i$ -th session of a party  $P$ , we write it as  $\pi_i^P$ . The index  $i$  does not necessarily represent the session’s order but rather serves as a unique identifier for that party’s session. In the security definition in Appendix A, we assume that  $i$  increases monotonically. A session  $\pi$  contains all necessary information for computing session keys or maintaining internal states in the session. More specifically, each session  $\pi$  is associated with the following variables. If a session  $\pi$  has not been initialized, then we have all variables set to  $\perp$ .

- `oid` and `pid` denote the identifiers of the owner and intended peer of the session, respectively.
- `role`  $\in \{\text{I}, \text{R}\}$  indicates the session is an initiator session or a responder session.
- `type`  $\in \{\text{full}, \text{reduced}\}$  indicates the session is run in the full mode or in the reduced mode. If the pre-key bundle of the session involves a one-time pre-key  $epk$ , then we have  $\pi.\text{type} = \text{full}$ ; Otherwise,  $\pi.\text{type} = \text{reduced}$ .
- `rand` stores the randomness used in the session.
- `ssk` stores the semi-static secret key used in the session. This variable is used when  $\pi$  is a responder session, and it is set to  $\perp$  if  $\pi$  is an initiator session.

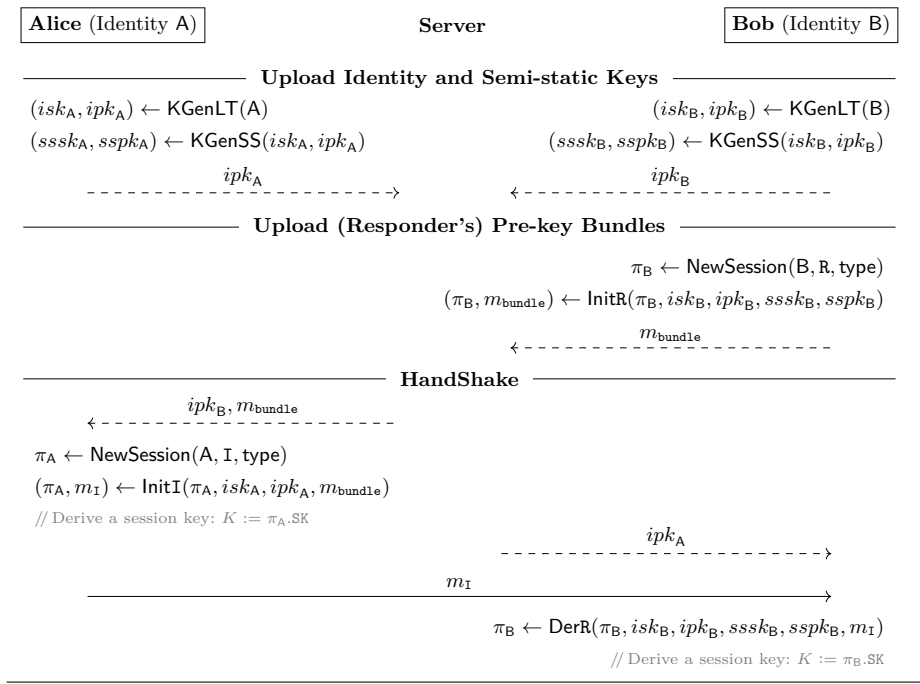


**Fig. 10.** Running a two-message initial handshake protocol between two parties.

- **sspk** records the semi-static public key used in the session.
- **cid** is the contributive identifier. In Signal’s X3DH setting, this variable simply stores the used pre-key bundle in the session.
- **sid** is the session identifier defining session partnering (or matching sessions).

The specific definition of **sid** may vary for different BAKE protocols. A good protocol design should ensure that the **sid** captures the *view* related to the session key derivation. For example, the view should include the session peer identifier, the used public keys, the sent and received protocol messages, and other information that may affect the session key derivation. This may help to prevent some attacks where an adversary tricks a session into deriving the same session key as another session with different peers or different protocol messages, e.g., the unknown key-share (UKS) attack [47].

- $SK \in \mathcal{SK}$  stores the session key. In this paper, if  $\pi.SK \neq \perp$ , then it indicates that  $\pi$  has completed successfully and accepted a session key.
- **state** stores the necessary state information for deriving session keys.

**BUNDLED AUTHENTICATED KEY EXCHANGE.** Let **P** be an arbitrary party, **A** be a session initiator, and **B** be a session responder. A bundled authenticated key exchange (BAKE) protocol includes the following algorithms:

- $\pi \leftarrow \text{NewSession}(P, \text{role}, \text{type})$ : The *session creation algorithm* initializes a session with session owner  $P$ , setting up the session's role (initiator or responder), type (full or reduced), and the randomness used in the session.
- $(isk_P, ipk_P) \leftarrow \text{KGenLT}(P)$ : The *identity (or long-term) key generation* algorithm generates an identity key pair for a party  $P$ .
- $(sssk_P, sspk_P) \leftarrow \text{KGenSS}(isk_P, ipk_P)$ : The *semi-static key generation* algorithm inputs an identity key pair and outputs a semi-static key pair.
- $(\pi, m_{\text{bundle}}) := \text{InitR}(\pi, isk, ipk, sssk, sspk)$ : The *responder initialization* algorithm (or *pre-key bundle generation* algorithm) takes a session  $\pi$ , an identity key pair, and a semi-static key pair as inputs, initializes  $\pi$  as a responder session, and outputs a pre-key bundle message  $m_{\text{bundle}}$ . The algorithm also updates the session variable  $\pi.\text{cid} := m_{\text{bundle}}$ .
- $(\pi, m_I) := \text{InitI}(\pi, isk, ipk, m_{\text{bundle}})$ : Upon receiving a pre-key bundle  $m_{\text{bundle}}$ , the *initiator initialization* algorithm inputs a session  $\pi$  and an initiator identity key pair, initializes  $\pi$  as an initiator session, and outputs an initiator protocol message  $m_I$ . The algorithm also updates the session variable  $\pi.\text{cid} := m_{\text{bundle}}$ . If  $\pi$  has been used, or  $m_{\text{bundle}}$  is invalid (e.g., the signature in the pre-key bundle cannot be verified), then  $\text{InitI}$  outputs  $\perp$ . Otherwise,  $\pi$  runs the protocol, derives a session key (i.e., updates  $\pi.\text{SK}$ ) and returns the initiator protocol message  $m_I$ .
- $\pi := \text{DerR}(\pi, isk, ipk, sssk, sspk, m_I)$ : Upon receiving the initiator protocol message  $m_I$ , the *responder key derivation* algorithm takes as inputs the responder session  $\pi$ , the responder's identity key pair, and semi-static key pair. If  $\pi$  has been used or  $m_I$  is invalid, then  $\text{DerR}$  returns  $\perp$ . Otherwise, it derives the session key, updates  $\pi.\text{SK}$ , and returns the updated session  $\pi$ .

Note that  $\text{InitR}$ ,  $\text{InitI}$ , and  $\text{DerR}$  are deterministic algorithms, because the internal randomness used in these algorithms is already included in the session variable  $\pi.\text{rand}$ , which is initialized by  $\text{NewSession}$ . Moreover, all algorithms have access to the list of public keys (e.g.,  $ipk$ ), which models the fact that parties can retrieve the peer's public keys from the server.

Figure 10 shows how two honest parties  $A$  and  $B$  run these algorithms to finish an initial handshake. The *correctness* definition requires that any honest execution of the initial handshake protocol between two parties produces the same session key. As both X3DH and XHMQV have perfect correctness, we omit the formal correctness definition here.

**DEFINING SECURITY.** Following prior work on secure messaging using initial handshake protocols, such as [23,24], we use a Bellare-Rogaway-style AKE security game to define the security of BAKE protocols. Roughly speaking, adversaries under this model are allowed to create sessions, send messages, and compromise some secret information (e.g., secret keys and randomness). The adversary's goal is to distinguish whether a challenge (namely, test) session key is real (i.e., derived from the protocol execution) or random. We call a BAKE protocol *secure* if no efficient adversary can achieve non-negligible distinguishing advantage, provided that the total secret information revealed across test sessions remains within the

model's restrictions. For readability, we present the complete security model in Appendix A.

### 3.2 BAKE Protocols: X3DH, X3DH<sup>+</sup>, and XHMQV

In this section, we analyze three protocols: X3DH, X3DH<sup>+</sup>, and XHMQV. X3DH is the original one from Signal's specification [45], while X3DH<sup>+</sup> is a variant of X3DH that includes the session context in the key derivation. XHMQV is a new protocol proposed very recently [24] that adapts the HMQV protocol [41] to Signal's setting. As a result, it achieves stronger security guarantees while requiring fewer group exponentiations.

Let  $\mathbb{G}$  be a group of prime order  $p$  with generator  $g$ . Let  $\text{KDF}_f$  and  $\text{KDF}_r : \{0,1\}^* \mapsto \mathcal{SK}$  be two key derivation functions for full sessions and reduced sessions, respectively. Let  $\Sigma$  be a signature scheme built over  $\mathbb{G}$ . For XHMQV, we further need four hash functions  $H_0, H_1, H_2$ , and  $H_3 : \{0,1\}^* \mapsto \mathbb{Z}_p$  to compute the coefficients for combining the keys. The description of these protocols is given in Figure 11.

**A BRIEF REVIEW ON DESIGN RATIONALE.** X3DH, X3DH<sup>+</sup>, and XHMQV have the same key pair structure. Let  $(a, A)$  and  $(x, X)$  be Alice's long-term (identity) and ephemeral key pairs, respectively. Similarly, let  $(b, B)$ ,  $(s, (S, \sigma))$ , and  $(y, Y)$  be Bob's long-term, semi-static (signed pre-), and ephemeral (one-time pre-) key pairs, respectively.

For convenience, we slightly abuse the notation: here we let  $\text{XX}_1\text{-XX}_2$  denote the combination of Alice's  $\text{XX}_1$  key pair and Bob's  $\text{XX}_2$  key pair in the protocol execution, where  $\text{XX}_1, \text{XX}_2 \in \{\text{LT}, \text{SS}, \text{Eph}\}$ . For example, LT-SS means the combination of Alice's long-term key and Bob's semi-static key, and Eph-Eph means the combination of Alice's ephemeral key and Bob's ephemeral key.

X3DH's key derivation is straightforward: it includes 3-4 combinations of plain Diffie-Hellman: LT-SS ( $g^{as}$ ), Eph-LT ( $g^{xb}$ ), Eph-SS ( $g^{xs}$ ), and Eph-Eph ( $g^{xy}$ , in the full mode). Similarly, X3DH<sup>+</sup> includes the same combinations as X3DH, but it also includes the session context in the key derivation (cf. the dashbox in Figure 11). All these combinations correspond to the attack scenarios defined in Figure 15. The impact of including the session context in the key derivation is discussed in Remark 2.

Instead, XHMQV's key derivation is more complicated. It combines all key pairs together (i.e.,  $g^{(x+da) \cdot (s+e_1b)}$  and  $g^{(x+da) \cdot (y+e_2s+e_1b)}$  in the full mode) following the design of HMQV, providing all possible combinations of the key pairs in different modes. XHMQV covers more attack scenarios and requires fewer group exponentiations. Like X3DH<sup>+</sup>, XHMQV also includes the session context in the key derivation, which is important for its security.

**SECURITY OF X3DH<sup>+</sup> AND XHMQV.** The security statements for X3DH<sup>+</sup> and XHMQV are presented in Theorems 1 and 2. For readability, we defer the proofs of Theorems 1 and 2 to Appendices B and C, respectively.

**Theorem 1 (Security of X3DH<sup>+</sup>).** *Let X3DH<sup>+</sup> be the BAKE protocol in Figure 11,  $\mathbb{G}$  be the underlying group of X3DH<sup>+</sup>, and  $\Sigma$  be the signature scheme*

| KGenLT(P)                                        | NewSession(P, role $\in \{I, R\}$ , type $\in \{\text{full}, \text{reduced}\}$ ) |                                                    |
|--------------------------------------------------|----------------------------------------------------------------------------------|----------------------------------------------------|
| 01 $isk_P := u \xleftarrow{\$} \mathbb{Z}_p$     | 09 Initialize a session $\pi$ :                                                  |                                                    |
| 02 $ipk_P := U := g^u$                           | 10 $r \xleftarrow{\$} \mathbb{Z}_p^*$ // Random coins for the session            |                                                    |
| 03 <b>return</b> ( $isk_P, ipk_P$ )              | 11 $\pi.(\text{oid}, \text{role}, \text{rand}) := (P, \text{role}, r)$           |                                                    |
| KGenSS( $isk_P = u$ )                            | 12 <b>if</b> role = R                                                            |                                                    |
| 04 $s \xleftarrow{\$} \mathbb{Z}_p$ , $S := g^s$ | 13 <b>if</b> type = reduced // No one-time pre-key in the reduced mode           |                                                    |
| 05 $\sigma \leftarrow \text{Sign}(u, S)$         | 14 $\pi.\text{rand} := \perp$                                                    |                                                    |
| 06 $sssk := s$                                   | 15 $\pi.\text{type} := \text{type}$ // Set the type of the responder session     |                                                    |
| 07 $sspk := (S, \sigma)$                         | // The type of an initiator session will be set when running InitI               |                                                    |
| 08 <b>return</b> ( $sssk, sspk$ )                | 16 <b>return</b> $\pi$                                                           |                                                    |
| InitR( $\pi, isk_B, ipk_B, sssk, sspk$ )         | InitI( $\pi, isk_A, ipk_A, m_{\text{bundle}}$ )                                  | DerR( $\pi, isk_B, ipk_B, sssk, sspk, m$ )         |
| // Generate a pre-key bundle                     | 29 ( $a, A$ ) := ( $isk_A, ipk_A$ )                                              | 63 $B := \pi.\text{oid}$                           |
| 17 ( $b, B$ ) := ( $isk_B, ipk_B$ )              | 30 ( $B, (S, \sigma), Y$ ) := $m_{\text{bundle}}$                                | 64 ( $b, B$ ) := ( $isk_B, ipk_B$ )                |
| 18 ( $s, (S, \sigma)$ ) := ( $sssk, sspk$ )      | 31 $B := ipk[B]$                                                                 | 65 $s := sssk$ , ( $S, \sigma$ ) := $sspk$         |
| 19 <b>if</b> $\pi.\text{type} = \text{full}$     | 32 <b>if</b> Verify( $B, \sigma, S$ ) $\neq \text{true}$                         | 66 ( $y, Y$ ) := $\pi.\text{state}$                |
| 20 $y := \pi.\text{rand}$ , $Y := g^y$           | 33 <b>return</b> $\perp$                                                         | 67 ( $A, X$ ) := $m$                               |
| 21 <b>else</b> $Y := \perp$                      | 34 <b>else</b> $\pi.sspk := (S, \sigma)$                                         | 68 $\text{ctxt}_r := (A, X, B, S, \sigma)$         |
| 22 $B := \pi.\text{oid}$                         | 35 <b>if</b> $Y \neq \perp$ : $\pi.\text{type} := \text{full}$                   | 69 $\text{ctxt}_t := (A, X, B, S, Y, \sigma)$      |
| 23 $m_{\text{bundle}} := (B, (S, \sigma), Y)$    | 36 <b>else</b> $\pi.\text{type} := \text{reduced}$                               | 70 // X3DH's key derivation                        |
| 24 Update $\pi$ :                                | 37 $x := \pi.\text{rand}$ , $X := g^x$                                           | 71 $DH := (A^s, X^b, X^s)$                         |
| 25 $\pi.\text{pid} := \star$                     | 38 // Session context                                                            | 72 <b>if</b> $\pi.\text{type} = \text{full}$       |
| // Unknown peer identity                         | 39 $\text{ctxt}_r := (A, X, B, S, \sigma)$                                       | 73 $DH := DH \parallel X^y$                        |
| 26 $\pi.\text{state} := (y, Y)$                  | 40 $\text{ctxt}_t := (A, X, B, S, Y, \sigma)$                                    | 74 // XHMQV's key derivation                       |
| 27 $\pi.\text{cid} := (B, (S, \sigma), Y)$       | 41 // X3DH's key derivation                                                      | 75 $d := H_0(A, X)$                                |
| 28 <b>return</b> ( $\pi, m_{\text{bundle}}$ )    | 42 $DH := (S^a, B^x, S^x)$                                                       | 76 <b>if</b> $\pi.\text{type} = \text{full}$       |
|                                                  | 43 <b>if</b> $\pi.\text{type} = \text{full}$                                     | 77 $e_1 := H_1(B, Y, S)$                           |
|                                                  | 44 $DH := DH \parallel Y^x$                                                      | 78 $e_2 := H_2(B, Y, S)$                           |
|                                                  | 45 // XHMQV's key derivation                                                     | 79 $DH := (XA^d)^{y+e_1b+e_2s}$                    |
|                                                  | 46 $d := H_0(A, X)$                                                              | 80 <b>else</b>                                     |
|                                                  | 47 <b>if</b> $\pi.\text{type} = \text{full}$                                     | 81 $e_1 := H_3(B, S)$                              |
|                                                  | 48 $e_1 := H_1(B, Y, S)$                                                         | 82 $DH := (XA^d)^{s+e_1b}$                         |
|                                                  | 49 $e_2 := H_2(B, Y, S)$                                                         | 83 <b>if</b> $\pi.\text{type} = \text{full}$       |
|                                                  | 50 $DH := (YB^{e_1}S^{e_2})^{x+da}$                                              | 84 $K := \text{KDF}_t(DH \parallel \text{ctxt}_t)$ |
|                                                  | 51 <b>else</b>                                                                   | 85 <b>else</b>                                     |
|                                                  | 52 $e_1 := H_3(B, S)$                                                            | 86 $K := \text{KDF}_t(DH \parallel \text{ctxt}_t)$ |
|                                                  | 53 $DH := (B^{e_1}S)^{x+da}$                                                     | 87 Update $\pi$ :                                  |
|                                                  | 54 <b>if</b> $\pi.\text{type} = \text{full}$                                     | 88 $\pi.(\text{pid}, \text{SK}) := (A, K)$         |
|                                                  | 55 $K := \text{KDF}_t(DH \parallel \text{ctxt}_t)$                               | 89 $\pi.\text{sid} := (A, X, B, S, Y, \sigma)$     |
|                                                  | 56 <b>else</b>                                                                   | 90 <b>return</b> $\pi$                             |
|                                                  | 57 $K := \text{KDF}_t(DH \parallel \text{ctxt}_t)$                               |                                                    |
|                                                  | 58 Update $\pi$ :                                                                |                                                    |
|                                                  | 59 $\pi.(\text{pid}, \text{SK}) := (B, K)$                                       |                                                    |
|                                                  | 60 $\pi.\text{cid} := (B, (S, \sigma), Y)$                                       |                                                    |
|                                                  | 61 $\pi.\text{sid} := (A, X, B, S, Y, \sigma)$                                   |                                                    |
|                                                  | 62 <b>return</b> ( $\pi, m := (A, X)$ )                                          |                                                    |

**Fig. 11.** Protocols X3DH<sup>+</sup> and XHMQV. Both X3DH and X3DH<sup>+</sup> ignore the gray part, while XHMQV ignores X3DH's DH derivation (cf. Lines 41 to 44 and Lines 70 to 73). The original X3DH does not include the session context in the key derivation, while X3DH<sup>+</sup> and XHMQV do.

in  $X3DH^+$ . For any adversary  $\mathcal{A}$  against  $X3DH^+$  in  $\text{BAKE}_{X3DH^+}^A$ , making  $Q_{LT}$  queries to  $\text{CORLT}$ ,  $Q_{SS}$  queries to  $\text{CORSS}$ ,  $N_{\text{ses}}$  queries to  $\text{SEND}$  and  $\text{DER}$ ,  $Q_{\text{Eph}}$  queries to  $\text{REVRAND}$ ,  $Q_{\text{test}}$  queries to  $\text{TEST}$ , and  $Q_{\text{revSK}}$  queries to  $\text{REVSK}$ , there exist an adversary  $\mathcal{F}$  against the  $\text{EUF}^c\text{-1CMA-DDH}$  security of  $\Sigma$  and an adversary  $\mathcal{B}_{\text{dh}}$  against the  $\text{CorrGapCDH}$  assumption such that  $\mathbf{T}(\mathcal{F}) \approx \mathbf{T}(\mathcal{A})$ ,  $\mathbf{T}(\mathcal{B}_{\text{dh}}) \approx \mathbf{T}(\mathcal{A})$ , and

$$\begin{aligned} \text{Adv}_{X3DH^+}^{\text{BAKE}}(\mathcal{A}) &\leq \text{Adv}_{N_{LT}, N_{SS}, Q_{LT}, 4Q_{\text{kdf}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{F}) + \text{Adv}_{N_{LT}+N_{SS}+N_{\text{ses}}, Q_{LT}+Q_{SS}+Q_{\text{Eph}}, 4Q_{\text{kdf}}}^{\text{CorrGapCDH}}(\mathcal{B}_{\text{dh}}) \\ &\quad + N_{LT} \cdot \delta_{\text{sim}} + \frac{(N_{LT} + N_{SS} + N_{\text{ses}})^2}{2p}, \end{aligned} \quad (2)$$

where  $N_{LT}$  is the number of parties,  $n_{SS}$  is the number of semi-static keys per party,  $N_{SS} := n_{SS} \cdot N_{LT}$  is the total number of semi-static keys, and  $Q_{\text{kdf}}$  is the total number of  $\text{KDF}_f$  and  $\text{KDF}_r$  queries.

**Theorem 2 (Security of XHMQV).** Let XHMQV be the BAKE protocol in Figure 11,  $\mathbb{G}$  be the underlying group of XHMQV, and  $\Sigma$  be the signature scheme in XHMQV. For any adversary  $\mathcal{A}$  against  $\text{BAKE}_{XHMQV}^A$ , which makes  $Q_{LT}$  queries to  $\text{CORLT}$ ,  $Q_{SS}$  queries to  $\text{CORSS}$ ,  $N_{\text{ses}}$  queries to  $\text{SEND}$  and  $\text{DER}$ ,  $Q_{\text{Eph}}$  queries to  $\text{REVRAND}$ ,  $Q_{\text{test}}$  queries to  $\text{TEST}$ , and  $Q_{\text{revSK}}$  queries to  $\text{REVSK}$ , there exist an adversary  $\mathcal{F}$  against the  $\text{EUF}^c\text{-1CMA-DDH}$  security of  $\Sigma$  and an adversary  $\mathcal{B}_{\text{dh}}$  against the  $\text{CorrCRGapCDH}$  assumption such that  $\mathbf{T}(\mathcal{F}) \approx \mathbf{T}(\mathcal{A})$ ,  $\mathbf{T}(\mathcal{B}_{\text{dh}}) \approx \mathbf{T}(\mathcal{A})$ , and

$$\begin{aligned} \text{Adv}_{XHMQV}^{\text{BAKE}}(\mathcal{A}) &\leq \text{Adv}_{N_{LT}, N_{SS}, Q_{LT}, Q_{\text{kdf}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{F}) \\ &\quad + 2\text{Adv}_{N_{LT}+N_{SS}+N_{\text{ses}}, Q_{LT}+Q_{SS}+Q_{\text{Eph}}+3, Q_{\text{kdf}}, Q_{\text{h}}+1}^{\text{CorrCRGapCDH}}(\mathcal{B}_{\text{dh}}) \\ &\quad + N_{LT} \cdot \delta_{\text{sim}} + \frac{(N_{LT} + N_{SS} + N_{\text{ses}})^2 + 2Q_{\text{h}}}{2p}, \end{aligned}$$

where  $N_{LT}$  is the number of parties,  $n_{SS}$  is the number of semi-static keys per party,  $N_{SS} := n_{SS} \cdot N_{LT}$  is the total number of semi-static keys,  $Q_{\text{kdf}}$  is the total number of  $\text{KDF}_f$  and  $\text{KDF}_r$  queries, and  $Q_{\text{h}}$  is the total number of  $H_0$ ,  $H_1$ ,  $H_2$ , and  $H_3$  queries.

We then apply our AGM/ABROM results in Figure 2 to get security bounds for  $X3DH^+$  and XHMQV under the DL assumption. For the sake of simplicity, we assume that  $(N_{LT} + N_{SS} + N_{\text{ses}}) = O(N_{\text{ses}})$  since the number of sessions usually dominates the number of parties and semi-static keys in practice for large-scale AKE protocols. Ignoring the statistically negligible terms, we obtain the following corollary for  $X3DH^+$  and XHMQV.

**Corollary 1.** With the notation and assumptions in Theorems 1 and 2, if we further assume that (1)  $\mathbb{G}$  is analyzed in the AGM and ABROM, and (2)  $\Sigma = \text{XEdDSA}$  (with underlying group  $\mathbb{G}$ ), then we have

$$\begin{aligned} \text{Adv}_{X3DH^+}^{\text{BAKE}}(\mathcal{A}) &\leq O((N_{LT} + N_{SS} + N_{\text{ses}})^2 Q_{\text{kdf}} \cdot \epsilon_{\text{DL}}), \\ \text{Adv}_{XHMQV}^{\text{BAKE}}(\mathcal{A}) &\leq O((N_{LT} + N_{SS} + N_{\text{ses}})^2 Q_{\text{kdf}} \cdot \epsilon_{\text{DL}}). \end{aligned}$$

*Remark 1 (Comparing with Prior Works).* In Table 1, we compare the security bounds of X3DH<sup>+</sup> and XHMQV with prior analyses on X3DH and XHMQV ([16,23,24]). To obtain a fair and clear comparison, we also use asymptotic notations to express the security bounds (ignoring the statistically negligible terms), and apply our AGM/ABROM results in Figure 2 to their bounds to get the security bounds under the DL assumption in a multi-test model (cf. the fourth column). The second column shows their original security bounds.

The assumptions used in these works are similar to ours but in the single-user, single-challenge setting. Moreover, their security models only allow single-test query. By applying a standard hybrid argument, we can get security bounds under the multi-test model, which incurs an extra  $Q_{\text{test}} \leq N_{\text{ses}}$  factor in the security loss. In the realistic setting, the TEST oracle actually provides an interface for some outer layer protocol (e.g., the subsequent double ratchet algorithm) to access the AKE session keys. Thus, we have  $Q_{\text{test}} \lesssim N_{\text{ses}}$  (i.e., the number of tested sessions is bounded by the number of sessions).

*Remark 2 (Including Session Context in the KDF and Its Impact).* Signal’s original design of X3DH protocol does not include the session context in the key derivation. This does not affect its security thanks to its combination of the DH secrets (which is unlike XHMQV that must include the context to exclude unknown-key-share attack [41]). However, including the session context in the key derivation brings several noticeable benefits. We discuss it below.

From a *theoretical* point of view, including the context enables tight and efficient reductions, as stated in Theorem 1, which provides a better theoretical security bound than prior analyses on X3DH (cf. Table 1). One may wonder if the same tightness can be achieved without including the context. However, as far as we know, the answer is no, at least when we consider *efficient* reductions. The reason is quite technical. Intuitively, without the context, the reduction cannot efficiently identify which target the adversary is attacking, and thus it either has to guess the target (which causes a linear loss in the number of targets, as in prior X3DH proofs), or it has to check which session the adversary is attacking every time the adversary makes a KDF query. The former approach causes extra security loss, while the latter approach causes extra (and extremely longer) running time of the reduction, which is not desirable.

From a *practical* point of view, since the group exponentiations dominate the running time of the protocol and hashing longer byte strings (i.e., including the context) only causes negligible overhead, including the session context in the key derivation does not cause noticeable performance degradation.

More importantly, without the session context, an adversary against X3DH may gain some potential advantage in practice. For example, when  $\mathcal{A}$  queries  $\text{KDF}_r$  with some  $(Z_1, Z_2, Z_3)$ , this query defines an *equivalence* class of sessions, i.e., all sessions that have the same  $Z_1$ ,  $Z_2$ , and  $Z_3$  values fall into the same class. By linear algebra facts, for fixed  $Z_1$ ,  $Z_2$ , and  $Z_3$ , there are many  $(A, X, B, S)$  tuples in the same class (3 equations with 4 unknown variables will lead to 1 free variable). Consequently, when  $\mathcal{A}$  queries  $\text{KDF}_r$  with some  $(Z_1, Z_2, Z_3)$ , it is actually attacking the entire class at once, rather than a specific session. This

is actually captured in the security loss (or inefficient reductions) we discussed above. In contrast, if the session context is included in the key derivation, then each KDF query can only attack at most one session, which provides stronger security guarantees in practice.

Therefore, since Signal is considering moving from classical to post-quantum handshake [22,33], which may include some major updates on the protocol design, we highly recommend including the session context in the key derivation of X3DH. Note that this discussion above is not only for secure messaging protocols, but also for general AKE protocols. Including session context appears to be good practice for general AKE protocols. For example, the standard TLS 1.3 protocol [57] also includes the session context in the key derivation via more complicated key schedule algorithms.

### 3.3 Extension to PQXDH

We discuss how our result may be extended to improve the security proof of PQXDH [42,23]. The PQXDH protocol is a hybrid protocol that combines the classical X3DH with a post-quantum KEM. Its main security goal is to achieve security against harvest-now-decrypt-later attacks. The key idea of PQXDH is to include a post-quantum KEM secret into the session key so that even if the classical DH secrets are compromised in the future, the session key remains secure. The full code of PQXDH and PQXDH<sup>+</sup> (namely, the variant of PQXDH with context in session key hashing) is presented in Figure 37.

Following Fiedler and Günther’s analysis on PQXDH, we consider two types of security for PQXDH<sup>+</sup>, post-quantum security and classic security. Essentially, the post-quantum security of PQXDH<sup>+</sup> is the same as the post-quantum security of PQXDH in [23, Theorem 5.2]. The classic security bound of PQXDH<sup>+</sup> can be improved using our techniques, compared to the classic security of PQXDH in [23, Theorem 5.1]. The classic security of PQXDH<sup>+</sup> is more involved, since PQXDH<sup>+</sup> has more cases to consider (which is the same as PQXDH).

More details of how to extend our result on X3DH<sup>+</sup> to improve PQXDH analyses are presented in Appendix F.

## 4 Variants of Diffie-Hellman Assumptions

We revisit the hardness of the multi-user CDH assumptions CorrGapCDH and CorrCRGapCDH [39], which are used in the security analyses of X3DH<sup>+</sup> (cf. Theorem 1) and XHMQV (cf. Theorem 2) respectively. The relevant definitions used in our proofs are given in Section 2.1.

Specifically, we prove that the Discrete Logarithm assumption implies these multi-user CDH assumptions with a loss factor of  $O(Q_{\text{CORR}}^2)$ , under the AGM and ABROM. Compared with the original analysis of Kiltz et al. in the Generic Group Model, our proofs allow adversaries to receive the group representation. Compared with their standard model analysis, our proofs do not rewind the adversary and thus offer tighter bounds. Moreover, we reduce the trivial  $O(n^2)$

guessing loss incurred in the reduction from the multiple-challenge setting with corruptions to the single-challenge setting to  $O(Q_{\text{CORR}}^2)$ .

By combining Lemmas 3 to 6, we can reduce the CorrGapCDH assumption to the DL assumption under the AGM and ABROM, as stated in the following Theorem 3.

**Theorem 3** ( $\text{DL} \xrightarrow[\text{AGM}+\text{ABROM}]{Q_{\text{CORR}}^2} \text{CorrGapCDH}$ ). *For any adversary  $\mathcal{A}$  against the CorrGapCDH assumption, making  $Q_{\text{DDH}}$  queries to the gap oracle DDH, there exists an algebraic adversary  $\mathcal{B}$  against the DL assumption such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{CorrGapCDH}}(\mathcal{A}) \leq e^2 \cdot (Q_{\text{CORR}} + 2)^2 \cdot \text{Adv}^{\text{DL}}(\mathcal{B}),$$

where  $n$  is the number of the input challenges, and  $0 < Q_{\text{CORR}} \leq n - 2$  is the number of corruption queries made by  $\mathcal{A}$ .

By combining Lemmas 1, 2, 5 and 6, we can reduce the CorrCRGapCDH assumption to the DL assumption under the AGM and ABROM, as stated in the following Theorem 4. To simplify the bound, here we use the fact that in Lemma 2, the number of queries to oracles BRO and  $\text{BRO}^{-1}$  is equal to  $Q_{\text{CH}}$ , which also corresponds to the number of times  $f$  is invoked in the reduction from the CorrCRGapCDH assumption to the  $f$ -GapCSqDH assumption in Lemma 1.

**Theorem 4** ( $\text{DL} \xrightarrow[\text{AGM}+\text{ABROM}]{Q_{\text{CORR}}^2} \text{CorrCRGapCDH}$ ). *For any adversary  $\mathcal{A}$  against the CorrCRGapCDH assumption, making  $Q_{\text{DDH}}$  queries and  $Q_{\text{CH}}$  queries to the gap oracle DDH and the challenge oracle CH, there exists an algebraic adversary  $\mathcal{B}$  against the DL assumption such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, Q_{\text{CH}}}^{\text{CorrCRGapCDH}}(\mathcal{A}) \leq e^2 \cdot \frac{(Q_{\text{CORR}} + 2)^2}{2} \cdot \text{Adv}^{\text{DL}}(\mathcal{B}) + \frac{O(n^4 + Q_{\text{CH}})}{p},$$

where  $n$  is the number of the input challenges, and  $0 < Q_{\text{CORR}} \leq n - 2$  is the number of corruption queries made by  $\mathcal{A}$ .

Comparing to the bound given in [39, Theorem 1], our bound gets rid of the quadratic loss  $\sqrt{\epsilon}$  and the loss factor  $Q_{\text{CH}}$  (the number of queries to the challenge oracle), and thus is much tighter. Moreover, we reduce the CorrCRGapCDH assumption to the DL assumption, whereas their work relies on the GapCDH assumption. Additionally, instead of a loss of  $O(n^2)$ , our reduction incurs only a loss of  $O(Q_{\text{CORR}}^2)$ . Since typically  $Q_{\text{CORR}} \leq n - 2$ , this yields a tighter reduction in practice.

**RELATIONS OF ASSUMPTIONS.** We present the following lemmas, which show the relations among several variants of the Diffie–Hellman assumptions, in order to establish Theorems 3 and 4. Among them, only Lemmas 1 and 3 are non-tight, incurring a loss factor of  $O(Q_{\text{CORR}}^2)$  due to corruptions, where  $Q_{\text{CORR}}$  denotes the number of corruptions. Lemmas 1, 3 and 4 are in the standard model, and Lemmas 5 and 6 are in the AGM. Lemma 2 is in the ABROM, which is a weaker

model than AGM. The definitions of these models are given in Section 2. For readability, we defer the proofs of these lemmas to Appendix D.

The following Lemma 1 shows that the CorrCRGapCDH assumption can be implied by our newly defined  $f$ -GapCSqDH in the standard model. The reduction will lose a factor of  $O(Q_{\text{CORR}}^2)$ .

**Lemma 1** ( $f$ -GapCSqDH  $\xrightarrow{Q_{\text{CORR}}^2}$  CorrCRGapCDH). *For any adversary  $\mathcal{A}$  against the CorrCRGapCDH assumption, making  $Q_{\text{DDH}}$  queries and  $Q_{\text{CH}}$  queries to the gap oracle DDH and the challenge oracle CH, there exists a  $f$ -GapCSqDH adversary  $\mathcal{B}$  making  $Q_{\text{DDH}}$  queries to the DDH oracle such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, Q_{\text{CH}}}^{\text{CorrCRGapCDH}}(\mathcal{A}) \leq e^2 \cdot \frac{(Q_{\text{CORR}} + 2)^2}{4} \cdot \text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{B}) + \frac{O(n^4 + Q_{\text{CH}})}{p},$$

where  $n$  is the number of the input challenges,  $0 < Q_{\text{CORR}} \leq n - 2$  is the number of corruption queries made by  $\mathcal{A}$ , and  $f : \mathbb{G} \rightarrow \mathbb{Z}_p$  is an efficiently computable function.

The following Lemma 2 presents a tight reduction from the  $f$ -GapCSqDH assumption to the GapSqdH assumption in the ABROM. We adapt the proof technique of [15, Theorem 5] to our assumptions, namely GapSqdH and  $f$ -GapCSqDH.

**Lemma 2** (GapSqdH  $\xrightarrow{\text{ABROM}}$   $f$ -GapCSqDH). *Let  $f$  be a function modeled as the composition  $f = \Psi \circ \Pi \circ \varphi$ , where  $\varphi : \mathbb{G} \rightarrow \{0, 1\}^L$  is a semi-injective function,  $\Pi : \{0, 1\}^L \rightarrow [2^L]$  is modeled as a pair of bijective random oracles (BRO, BRO $^{-1}$ ), and  $\Psi : [2^L] \rightarrow \mathbb{Z}_p$  is an arbitrary efficiently computable function. For any  $f$ -GapCSqDH adversary  $\mathcal{A}$  making  $Q_{\text{DDH}}$  queries to the gap oracle DDH in the ABROM, there exists a GapSqdH adversary  $\mathcal{B}$  making  $Q_{\text{DDH}}$  queries to the DDH oracle such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{A}) \leq \text{Adv}_{Q_{\text{DDH}}}^{\text{GapSqdH}}(\mathcal{B}) + \frac{q + (q + 1) \cdot \text{MAX}_{\Psi}}{p},$$

where  $q$  is the total number of queries to BRO and BRO $^{-1}$ , and  $\text{MAX}_{\Psi} := \max_{x \in \mathbb{Z}_p} |\Psi^{-1}(x)|$  denotes the maximum preimage size of  $\Psi$ .

The following lemma is an improvement of Lemma 2 in [39]. Instead of losing  $n$ , ours loses only  $Q_{\text{CORR}}$ , where  $n$  is the maximum number of public keys and  $Q_{\text{CORR}}$  is the number of CORR queries. Usually,  $n > Q_{\text{CORR}}$ . Our technique adapts that of [5]. For readability, we defer the proof of Lemma 3 to Appendix D.4.

**Lemma 3** (GapCDH  $\xrightarrow{Q_{\text{CORR}}^2}$  CorrGapCDH). *For any adversary  $\mathcal{A}$  against the CorrGapCDH assumption, there exists an adversary  $\mathcal{B}$  against the GapCDH assumption such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{CorrGapCDH}}(\mathcal{A}) \leq \frac{e^2 \cdot (Q_{\text{CORR}} + 2)^2}{2} \cdot \text{Adv}_{Q_{\text{DDH}}}^{\text{GapCDH}}(\mathcal{B}),$$

where  $0 < Q_{\text{CORR}} \leq n - 2$  is the number of corruption queries made by  $\mathcal{A}$ , and all the notations are the same as in Definition 6.

Then, the following Lemma 4 shows that the GapSqDH assumption implies the GapCDH assumption in the standard model. We prove this by using the self-reducibility to give a tight reduction.

**Lemma 4 (GapSqDH  $\Rightarrow$  GapCDH).** *For any adversary  $\mathcal{A}$  against the GapCDH assumption, there exists an adversary  $\mathcal{B}$  against the GapSqDH assumption such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{Q_{\text{DDH}}}^{\text{GapCDH}}(\mathcal{A}) \leq \text{Adv}_{Q_{\text{DDH}}}^{\text{GapSqDH}}(\mathcal{B}),$$

where  $Q_{\text{DDH}}$  is the number of queries to the gap oracle DDH.

Although [27, Theorem 3.3] shows how to get rid of the gap oracle DDH in the AGM, by giving a reduction from the Strong Diffie-Hellman assumption [1] to the DL assumption with loss of  $4Q_{\text{DDH}}$ , their definition of the gap oracle is different from ours and the reduction is not tight. In particular, their gap oracle only accepts queries of the form  $(Y_\ell, Z_\ell)$ , and checks whether  $Z_\ell = Y_\ell^{\text{DLOG}_g(X)}$ , where  $X$  is one of the input challenges. In contrast, our gap oracle additionally allows the adversary to query with any arbitrary chosen  $X_\ell$  together, and checks whether  $Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)}$ . Consequently, their results cannot be directly applied in our setting.

Hence, we give a direct reduction in Appendix D.6 from the GapSqDH assumption to the SqDH and DL assumptions, as stated in Lemma 5.

**Lemma 5 (SqDH+DL  $\xrightarrow{\text{AGM}}$  GapSqDH).** *For any algebraic adversary  $\mathcal{A}$  against the GapSqDH assumption, making  $Q_{\text{DDH}}$  queries to the gap oracle DDH, there exists a DL adversary  $\mathcal{B}$  and a SqDH adversary  $\mathcal{C}$  such that  $\mathbf{T}(\mathcal{C}) \approx \mathbf{T}(\mathcal{A})$ ,  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{Q_{\text{DDH}}}^{\text{GapSqDH}}(\mathcal{A}) \leq \text{Adv}^{\text{DL}}(\mathcal{B}) + \text{Adv}^{\text{SqDH}}(\mathcal{C}).$$

**Lemma 6 (DL  $\xrightarrow{\text{AGM}}$  SqDH [27, Theorem 3.1]).** *The Discrete Logarithm assumption is tightly equivalent to the square Diffie-Hellman assumption in the Algebraic Group Model. That is, for any algebraic adversary  $\mathcal{A}$  against the SqDH assumption, there exists an algebraic adversary  $\mathcal{B}$  against the DL assumption such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}^{\text{SqDH}}(\mathcal{A}) = \text{Adv}^{\text{DL}}(\mathcal{B}).$$

## 5 Digital Signatures

We prove the security of two signature schemes, XEdDSA and ECDSA, where the former is used in X3DH. In order to obtain the security of X3DH<sup>+</sup> (cf. Theorem 1) and XHMQV (cf. Theorem 2), we rely on the EUF<sup>c</sup>-1CMA-DDH security (cf. Definition 8) of the signature scheme.

| $\text{KG}(\lambda)$                | $\text{Sign}(x, m)$                                   | $\text{Verify}(X, m, \sigma)$                              |
|-------------------------------------|-------------------------------------------------------|------------------------------------------------------------|
| 01 $x \xleftarrow{\$} \mathbb{Z}_p$ | 05 $\text{INP} \xleftarrow{\$} \{0, 1\}^{\text{len}}$ | 11 parse $\sigma$ as $(R, s)$                              |
| 02 $X := g^x$                       | 06 $r := G(x \  m \  \text{INP})$                     | 12 $h := H(R \  X \  m)$                                   |
| 03 $vk := X; sk := x$               | 07 $R := g^r$                                         | 13 <b>return</b> $\llbracket g^s = R \cdot X^h \rrbracket$ |
| 04 <b>return</b> $(vk, sk)$         | 08 $h := H(R \  X \  m)$                              |                                                            |
|                                     | 09 $s := r + h \cdot x$                               |                                                            |
|                                     | 10 <b>return</b> $\sigma := (R, s)$                   |                                                            |

**Fig. 12.** The XEdDSA signature scheme in Definition 10

We give a tight reduction from the  $\text{EUF}^c\text{-1CMA-DDH}$  security of XEdDSA to the  $f\text{-CorrGapCDL}$  assumption (cf. Definition 5) in the standard model. We then analyze the hardness of  $f\text{-CorrGapCDL}$  in the generic group model. Furthermore, under the ABROM and AGM, we prove that  $f\text{-CorrGapCDL}$  can be reduced to the DL assumption with a loss factor of  $2e(Q_{\text{CORR}} + 1)$ , where  $Q_{\text{CORR}}$  is the number of corruption queries. Using a similar technique, we give a tight reduction from the  $\text{EUF-CMA}$  security of ECDSA to the DL assumption in the ABROM. For readability, we defer the proofs of the results in this section to Appendix E.

Let  $\lambda$  be the security parameter. On input  $\lambda$ , the setup algorithm runs a parameter-generation algorithm associated with the signature scheme to obtain an elliptic curve  $E$  over a finite field  $\mathbb{F}_{p'}$ , a cyclic subgroup  $\mathbb{G} \subseteq E(\mathbb{F}_{p'})$  of prime order  $p$ , and a generator  $g$ . It outputs the public parameters  $(\mathbb{G}, g, p)$ .

### 5.1 Security of XEdDSA

**Definition 10 (XEdDSA [54]).** Let  $\mathcal{M}$  be the message space, and  $\text{INP}$  denotes the 64-byte secure random data as part of the signing input. Let  $G$  and  $H$  be hash functions mapping into  $\mathbb{Z}_p$ . The XEdDSA signature scheme is given in Figure 12.

We follow the proof technique of [15, Theorem 1] and give the following Theorem 5. Note that our proof can be directly extended to the  $\text{EUF-CMA}$  notion under the multi-user setting by removing the signing-query restriction in Line 06, without compromising the proof. Our reduction is tight, as it reduces to a multi-variant assumption with corruptions.

**Theorem 5 ( $\text{EUF}^c\text{-1CMA-DDH}$  of XEdDSA).** For any adversary  $\mathcal{A}$  against the  $\text{EUF}^c\text{-1CMA-DDH}$  security of the XEdDSA, making  $Q_S$  queries to the signing oracle  $\text{SIGN}$ ,  $Q_{\text{DDH}}$  queries to the gap oracle  $\text{DDH}$ , and  $Q_G$  and  $Q_H$  queries to the random oracles  $G$  and  $H$ , respectively, there exists a  $f\text{-CorrGapCDL}$  adversary  $\mathcal{B}$  making  $Q_{\text{CORR}}$  queries to the  $\text{CORR}$  oracle and  $Q_{\text{DDH}}$  queries to the  $\text{DDH}$  oracle such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and

$$\begin{aligned} \text{Adv}_{n, Q_S, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{A}) &\leq \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{f\text{-CorrGapCDL}}(\mathcal{B}) \\ &\quad + \frac{Q_S \cdot (2Q_G + Q_H) + Q_H \cdot |f^{-1}(0)| + 1}{p} + \frac{Q_S \cdot Q_G}{2^{\text{len}}}, \end{aligned}$$

where  $n$  is the number of users and  $f : \mathbb{G} \rightarrow \mathbb{Z}_p$  is an efficient function.

Compared to the security analysis of XEdDSA in [12], we avoid the loss factor  $Q_H$  and the square-root loss  $\sqrt{\epsilon}$  by using the circular assumption technique. Note that this step is carried out solely in the random oracle model.

In the following discussions, we analyze the hardness of the  $f$ -CorrGapCDL assumption in the generic group model and the algebraic group model. The corresponding results are given in Theorem 6 and Corollary 2.

**$f$ -CorrGapCDL in the EC-GGM** In this part, we use the elliptic-curve generic group model (EC-GGM) [29] to analyze the hardness of  $f$ -CorrGapCDL. The EC-GGM is a commonly used model for constructions defined over elliptic-curve groups, as it captures certain non-generic features of ECDSA, including the conversion function and malleability. As argued in [15], this model is considered more realistic than the original GGM, in which group elements are encoded as random points on an elliptic curve  $E$  of prime order  $p$ , rather than as random strings.

EC-GGM. In the EC-GGM, the challenger uniformly chooses an injective encoding function  $\tau : \mathbb{Z}_p \rightarrow E$ , where  $\tau(0)$  is the identity element of  $E$ , and for every  $i \in \mathbb{Z}_p$ , if  $\tau(i) = P$ , then  $\tau(-i) = P^{-1}$ . The adversary queries the oracle ADD to compute group operations. Note that in our simulation, the ADD oracle includes the simulation of the MAP oracle in [15], since a query to MAP on input  $\ell \in \mathbb{Z}_p$  is equivalent to querying  $\text{ADD}(\ell, 1_E, 0, 1_E)$ .

**Theorem 6 (Generic Hardness of  $f$ -CorrGapCDL).** *Let  $E$  be an elliptic curve of prime order  $p$  and  $f : E \rightarrow \mathbb{Z}_p$  be a function. For an adversary  $\mathcal{A}$  against the  $f$ -CorrGapCDL assumption in the EC-GGM, making  $Q_{\text{ADD}}$  queries to the group operation oracle ADD,  $Q_{\text{CORR}}$  and  $Q_{\text{DDH}}$  queries to the corruption oracle CORR and the gap oracle DDH, respectively,  $\mathcal{A}$ 's advantage is*

$$\begin{aligned} \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, \text{EC-GGM}}^{f\text{-CorrGapCDL}}(\mathcal{A}) \\ \leq \frac{(2Q_{\text{ADD}} + 2n + 1)^2 + (n - Q_{\text{CORR}}) \cdot \text{MAX}_f + 2Q_{\text{DDH}} + 1}{p - 1}, \end{aligned} \quad (3)$$

where  $n$  is the number of input challenges.

Note that when  $f$  is chosen as the ECDSA conversion function, we have  $\text{MAX}_f = 2$ . For readability, we defer the proof of Theorem 6 to Appendix E.2.

**$f$ -CorrGapCDL in the ABROM and AGM** In this part, we show that  $f$ -CorrGapCDL can be reduced to DL with a loss factor of  $e \cdot (Q_{\text{CORR}} + 1)$ , where  $Q_{\text{CORR}}$  is the number of corruption queries in  $f$ -CorrGapCDL, using algebraic structures including ABROM and the AGM (cf. Corollary 2). The loss arises from corruptions in the multi-instance setting in the standard model and cannot be avoided by self-randomization.

Applying [5, Theorem 4.2] to  $f$ -CorrGapCDL and using the self-reducibility of  $f$ -GapCDL, we obtain the following Lemma 7.

**Lemma 7** ( $f\text{-GapCDL} \xrightarrow{Q_{\text{CORR}}} f\text{-CorrGapCDL}$ ). *Let  $f : \mathbb{G} \rightarrow \mathbb{Z}_p$  be an efficient function. For any adversary  $\mathcal{A}$  against the  $f\text{-CorrGapCDL}$  assumption, making  $Q_{\text{DDH}}$  queries to the gap oracle DDH, there exists a  $f\text{-GapCDL}$  adversary  $\mathcal{B}$  making  $Q_{\text{DDH}}$  queries to the DDH oracle such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{f\text{-CorrGapCDL}}(\mathcal{A}) \leq e(Q_{\text{CORR}} + 1) \cdot \text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCDL}}(\mathcal{B}),$$

where  $n$  is the number of the input challenges, and  $0 \leq Q_{\text{CORR}} < n$  is the number of corruption queries.

We adapt the result from [15, Theorem 5] and present the following Lemma 8. The only difference is that, in our setting, both assumptions are additionally given access to the same gap oracle DDH, which does not affect the proof and the conclusion.

**Lemma 8** ( $\text{GapDL} \xrightarrow{\text{ABROM}} f\text{-GapCDL}$ ). *Let  $L$  be an integer such that  $p \leq 2^L$ . Let  $f$  be a function modeled as the composition  $f = \Psi \circ \Pi \circ \varphi$ , where  $\varphi : \mathbb{G} \rightarrow \{0, 1\}^L$  is a semi-injective function,  $\Pi : \{0, 1\}^L \rightarrow [2^L]$  is modeled as a bijective random oracle, and  $\Psi : [2^L] \rightarrow \mathbb{Z}_p$  is an arbitrary efficiently computable function. For any  $f\text{-GapCDL}$  adversary  $\mathcal{A}$  making  $Q_{\text{DDH}}$  queries to the gap oracle DDH, there exists a  $\text{GapDL}$  adversary  $\mathcal{B}$  making  $Q_{\text{DDH}}$  queries to the DDH oracle such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCDL}}(\mathcal{A}) \leq \text{Adv}_{Q_{\text{DDH}}}^{\text{GapDL}}(\mathcal{B}) + \frac{(q + 1) \cdot \text{MAX}_{\Psi} + q^2 + 2q}{p},$$

where  $q$  is the total number of queries to oracles BRO and  $\text{BRO}^{-1}$  when  $\mathcal{A}$  computes the conversion function  $f$  or its inverse.

**Lemma 9** ( $\text{DL} \xrightarrow{\text{AGM}} \text{GapDL}$ ). *For any algebraic adversary  $\mathcal{A}$  against the  $\text{GapDL}$  assumption, making  $Q_{\text{DDH}}$  queries to the gap oracle DDH, there exists a DL adversary  $\mathcal{B}$  such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{Q_{\text{DDH}}}^{\text{GapDL}}(\mathcal{A}) \leq 2 \cdot \text{Adv}^{\text{DL}}(\mathcal{B}).$$

The following corollary is obtained by applying Lemmas 7 to 9.

**Corollary 2** ( $\text{DL} \xrightarrow[\text{AGM} + \text{ABROM}]{Q_{\text{CORR}}} f\text{-CorrGapCDL}$ ). *Let  $f : \mathbb{G} \rightarrow \mathbb{Z}_p$  be an efficient function. For any adversary  $\mathcal{A}$  against the  $f\text{-CorrGapCDL}$  assumption, making  $Q_{\text{DDH}}$  queries to the gap oracle DDH, there exists an algebraic adversary  $\mathcal{B}$  against the DL assumption such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\begin{aligned} \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{f\text{-CorrGapCDL}}(\mathcal{A}) &\leq 2e(Q_{\text{CORR}} + 1) \cdot \text{Adv}^{\text{DL}}(\mathcal{B}) \\ &\quad + \frac{e(Q_{\text{CORR}} + 1)((q + 1) \cdot \text{MAX}_{\Psi} + q^2 + 2q)}{p}. \end{aligned}$$

where  $n$  is the number of the input challenges,  $0 \leq Q_{\text{CORR}} < n$  is the number of corruption queries in  $f\text{-CorrGapCDL}$ , and  $q$  is the number of queries to  $f$  in the ABROM, corresponding to  $Q_H$  (i.e., the number of random oracle queries to  $H$ ) in Theorem 5.

## 5.2 Security of ECDSA

**Definition 11 (ECDSA [37]).** Let  $\mathcal{M}$  be the message space and  $H : \mathcal{M} \rightarrow \mathbb{Z}_p$  be a hash function. Let  $f : \mathbb{G} \rightarrow \mathbb{Z}_p$  be a conversion function that maps a point  $A$  to its  $x$ -coordinate interpreted as an integer in  $\mathbb{Z}_p$ . The Elliptic Curve Digital Signature Algorithm (ECDSA) signature scheme is given in Figure 13.

| $\text{KG}(\lambda)$                | $\text{SIGN}(x, m)$                 | $\text{VER}(X, m, \sigma)$                        |
|-------------------------------------|-------------------------------------|---------------------------------------------------|
| 01 $x \xleftarrow{\$} \mathbb{Z}_p$ | 05 $k \xleftarrow{\$} \mathbb{Z}_p$ | 10 parse $\sigma$ as $(r, s)$                     |
| 02 $X := g^x$                       | 06 $R := g^k$                       | 11 $R := g^{H(m)/s} X^{r/s}$                      |
| 03 $vk := X; sk := x$               | 07 $r \leftarrow f(R)$              | 12 <b>return</b> $\llbracket f(R) = r \rrbracket$ |
| 04 <b>return</b> $(vk, sk)$         | 08 $s := k^{-1} \cdot (H(m) + xr)$  |                                                   |
|                                     | 09 <b>return</b> $\sigma := (r, s)$ |                                                   |

**Fig. 13.** The ECDSA signature scheme in Definition 11

Following the brief sketch in [56, Section 1.4], which only shortly outlines how to recover  $x$ , we prove the EUF-CMA security of the ECDSA in the ABROM under the DL assumption, as shown in Theorem 7. In particular, we explicitly analyze the conditions under which  $x$  can be solved (i.e., ensuring the denominators in the equations are non-zero).

For readability, we defer the proof of Theorem 7 to Appendix E.4.

**Theorem 7 (Security of ECDSA, ABROM).** Let  $L$  be an integer such that  $p \leq 2^L$ . Let  $f = \Psi \circ \Pi \circ \varphi$  be the ECDSA conversion function, where  $\varphi : \mathbb{G} \rightarrow \{0, 1\}^L$  is a semi-injective function,  $\Pi : \{0, 1\}^L \rightarrow [2^L]$  is modeled as a bijective random oracle, and  $\Psi : [2^L] \rightarrow \mathbb{Z}_p$  is an arbitrary efficiently computable function. For any adversary  $\mathcal{A}$  against the EUF-CMA security of the ECDSA, making  $Q_S$  queries to the signing oracle SIGN,  $Q_H$  queries to the random oracle  $H$ , and  $q$  queries to oracles BRO and  $\text{BRO}^{-1}$  when computing the conversion function  $f$  or its inverse, there exists a DL adversary  $\mathcal{B}$  such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and

$$\text{Adv}_{Q_S, Q_H}^{\text{EUF-CMA}}(\mathcal{A}) \leq \text{Adv}^{\text{DL}}(\mathcal{B}) + \frac{2q + (q + 1) \cdot \text{MAX}_{\Psi} + Q_H^2}{p}.$$

**Acknowledgments.** We thank the anonymous reviewers of Asiacrypt 2026 for their valuable comments and for suggesting that we should discuss how to extend our techniques to improve the security analysis of PQXDH.

We used ChatGPT and OpenAI Codex for manuscript editing and consistency checks. ChatGPT suggested corrections, and we implemented them either ourselves or with assistance from Codex. We reviewed and double-checked the changes made by Codex. Further details are provided in Appendix G. We take responsibility for the final content.

Jiawei Bao and Jiaxin Pan were funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 572525990.

## References

1. Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Berlin, Heidelberg (Apr 2001) [9](#), [24](#)
2. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Berlin, Heidelberg (Apr / May 2002) [4](#)
3. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Berlin, Heidelberg (May 2000) [3](#)
4. Bellare, M., Palacio, A.: GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer, Berlin, Heidelberg (Aug 2002) [7](#)
5. Bellare, M., Riepel, D., Tessaro, S., Zhang, Y.: Count corruptions, not users: Improved tightness for signatures, encryption and authenticated key exchange. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part II. LNCS, vol. 15485, pp. 326–360. Springer, Singapore (Dec 2024) [7](#), [23](#), [26](#), [54](#), [61](#)
6. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93. pp. 62–73. ACM Press (Nov 1993) [6](#)
7. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO’93. LNCS, vol. 773, pp. 232–249. Springer, Berlin, Heidelberg (Aug 1994) [33](#)
8. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Berlin, Heidelberg (May / Jun 2006) [8](#)
9. Benedikt, B.J., Clermont, S., Fischlin, M., Schmalz, T.: Efficiency improvements for signal’s handshake protocol. Cryptology ePrint Archive, Paper 2025/1843 (2025), <https://eprint.iacr.org/2025/1843>, Accepted by ASIACCS 2026 [8](#)
10. Bernstein, D.J., Duif, N., Lange, T., Schwabe, P., Yang, B.Y.: High-speed high-security signatures. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 124–142. Springer, Berlin, Heidelberg (Sep / Oct 2011) [8](#)
11. Bhargavan, K., Jacomme, C., Kiefer, F., Schmidt, R.: Formal verification of the PQXDH post-quantum key agreement protocol for end-to-end secure messaging. In: Balzarotti, D., Xu, W. (eds.) USENIX Security 2024. USENIX Association (Aug 2024), <https://www.usenix.org/conference/usenixsecurity24/presentation/bhargavan> [3](#)
12. Brendel, J., Cremers, C., Jackson, D., Zhao, M.: The provable security of Ed25519: Theory and practice. In: 2021 IEEE Symposium on Security and Privacy. pp. 1659–1676. IEEE Computer Society Press (May 2021) [4](#), [8](#), [26](#)
13. Brendel, J., Fiedler, R., Günther, F., Janson, C., Stebila, D.: Post-quantum asynchronous deniable key exchange and the Signal handshake. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part II. LNCS, vol. 13178, pp. 3–34. Springer, Cham (Mar 2022) [3](#)
14. Chatterjee, S., Kobitz, N., Menezes, A., Sarkar, P.: Another look at tightness II: Practical issues in cryptography. Cryptology ePrint Archive, Report 2016/360 (2016), <https://eprint.iacr.org/2016/360> [7](#)

15. Cho, G., Fuchsbauer, G., O'Neill, A., Sefranek, M.: Schnorr signatures are tightly secure in the ROM under a non-interactive assumption. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part VI. LNCS, vol. 16005, pp. 223–255. Springer, Cham (Aug 2025) [6](#), [9](#), [10](#), [11](#), [23](#), [25](#), [26](#), [27](#)
16. Cohn-Gordon, K., Cremers, C., Dowling, B., Garratt, L., Stebila, D.: A formal security analysis of the Signal messaging protocol. In: 2017 IEEE European Symposium on Security and Privacy. pp. 451–466. IEEE Computer Society Press (Apr 2017) [3](#), [4](#), [5](#), [7](#), [20](#), [33](#)
17. Cohn-Gordon, K., Cremers, C., Dowling, B., Garratt, L., Stebila, D.: A formal security analysis of the Signal messaging protocol. *Journal of Cryptology* 33(4), 1914–1983 (Oct 2020) [3](#), [4](#), [5](#)
18. Cohn-Gordon, K., Cremers, C., Gjosteen, K., Jacobsen, H., Jager, T.: Highly efficient key exchange protocols with optimal tightness. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 767–797. Springer, Cham (Aug 2019) [3](#)
19. Collins, D., Huguenin-Dumittan, L., Nguyen, N.K., Rolin, N., Vaudenay, S.: K-waay: Fast and deniable post-quantum X3DH without ring signatures. In: Balzarotti, D., Xu, W. (eds.) USENIX Security 2024. USENIX Association (Aug 2024), <https://www.usenix.org/conference/usenixsecurity24/presentation/collins> [3](#)
20. Degabriele, J.P., Lehmann, A., Paterson, K.G., Smart, N.P., Strefer, M.: On the joint security of encryption and signature in EMV. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 116–135. Springer, Berlin, Heidelberg (Feb / Mar 2012) [4](#)
21. Dobson, S., Galbraith, S.D.: Post-quantum Signal key agreement from SIDH. In: Cheon, J.H., Johansson, T. (eds.) Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022. pp. 422–450. Springer, Cham (Sep 2022) [3](#)
22. Dodis, Y., Jost, D., Katsumata, S., Prest, T., Schmidt, R.: Triple ratchet: A bandwidth efficient hybrid-secure signal protocol. In: Fehr, S., Fouque, P.A. (eds.) EUROCRYPT 2025, Part VIII. LNCS, vol. 15608, pp. 302–331. Springer, Cham (May 2025) [21](#)
23. Fiedler, R., Günther, F.: Security analysis of Signal’s PQXDH handshake. In: Jager, T., Pan, J. (eds.) PKC 2025, Part II. LNCS, vol. 15675, pp. 137–169. Springer, Cham (May 2025) [3](#), [4](#), [7](#), [8](#), [14](#), [16](#), [20](#), [21](#), [34](#), [37](#), [73](#), [75](#)
24. Fiedler, R., Günther, F., Pan, J., Zeng, R.: XHMQV: Better efficiency and stronger security for signal’s initial handshake based on HMQV. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part VIII. LNCS, vol. 16007, pp. 109–140. Springer, Cham (Aug 2025) [2](#), [3](#), [4](#), [5](#), [6](#), [8](#), [9](#), [12](#), [13](#), [14](#), [16](#), [17](#), [20](#), [37](#)
25. Fiedler, R., Janson, C.: A deniability analysis of Signal’s initial handshake PQXDH. *PoPETs* 2024(4), 907–928 (Oct 2024) [3](#)
26. Fisher, R.A., Yates, F.: *Statistical Tables for Biological, Agricultural and Medical Research*. Oliver and Boyd, Edinburgh (1938) [61](#)
27. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Cham (Aug 2018) [6](#), [9](#), [24](#)
28. Gay, R., Hofheinz, D., Kohl, L., Pan, J.: More efficient (almost) tightly secure structure-preserving signatures. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 230–258. Springer, Cham (Apr / May 2018) [3](#)
29. Groth, J., Shoup, V.: On the security of ECDSA with additive key derivation and presignatures. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part I. LNCS, vol. 13275, pp. 365–396. Springer, Cham (May / Jun 2022) [26](#)

30. Haber, S., Pinkas, B.: Securely combining public-key cryptosystems. In: Reiter, M.K., Samarati, P. (eds.) ACM CCS 2001. pp. 215–224. ACM Press (Nov 2001) 4
31. Hashimoto, K., Katsumata, S., Kwiatkowski, K., Prest, T.: An efficient and generic construction for Signal’s handshake (X3DH): Post-quantum, state leakage secure, and deniable. In: Garay, J. (ed.) PKC 2021, Part II. LNCS, vol. 12711, pp. 410–440. Springer, Cham (May 2021) 3
32. Hashimoto, K., Katsumata, S., Kwiatkowski, K., Prest, T.: An efficient and generic construction for Signal’s handshake (X3DH): Post-quantum, state leakage secure, and deniable. *Journal of Cryptology* 35(3), 17 (Jul 2022) 3
33. Hashimoto, K., Katsumata, S., Wiggers, T.: Bundled authenticated key exchange: A concrete treatment of signal’s handshake protocol and post-quantum security. In: Bauer, L., Pellegrino, G. (eds.) USENIX Security 2025. pp. 6777–6796. USENIX Association (Aug 2025), <https://www.usenix.org/conference/usenixsecurity25/presentation/hashimoto-key-exchange> 3, 14, 21
34. Hofheinz, D.: Adaptive partitioning. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 489–518. Springer, Cham (Apr / May 2017) 3
35. Jaeger, J., Mohan, D.I.: Generic and algebraic computation models: When AGM proofs transfer to the GGM. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part V. LNCS, vol. 14924, pp. 14–45. Springer, Cham (Aug 2024) 9
36. Jager, T., Kiltz, E., Riepel, D., Schäge, S.: Tightly-secure authenticated key exchange, revisited. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 117–146. Springer, Cham (Oct 2021) 3, 33
37. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security* 1, 36–63 (2001) 28
38. Josefsson, S., Liusvaara, I.: Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032 (Jan 2017), <https://www.rfc-editor.org/info/rfc8032> 8
39. Kiltz, E., Pan, J., Riepel, D., Ringerud, M.: Multi-user CDH problems and the concrete security of NAXOS and HMQV. In: Rosulek, M. (ed.) CT-RSA 2023. LNCS, vol. 13871, pp. 645–671. Springer, Cham (Apr 2023) 5, 6, 7, 8, 9, 11, 21, 22, 23, 61, 68
40. Kobeissi, N., Bhargavan, K., Blanchet, B.: Automated verification for secure messaging protocols and their implementations: A symbolic and computational approach. In: 2017 IEEE European Symposium on Security and Privacy. pp. 435–450. IEEE Computer Society Press (Apr 2017) 3
41. Krawczyk, H.: HMQV: A high-performance secure Diffie-Hellman protocol. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 546–566. Springer, Berlin, Heidelberg (Aug 2005) 8, 17, 20
42. Kret, E., Schmidt, R.: The PQXDH key agreement protocol. <https://signal.org/docs/specifications/pqxdh/> (2023), revision 3, last updated January 23, 2024 21, 73
43. LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger security of authenticated key exchange. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 1–16. Springer, Berlin, Heidelberg (Nov 2007) 8
44. Loss, J.: New techniques for the modular analysis of digital signature schemes. Ph.D. thesis, Ruhr University Bochum, Germany (2019), <https://hss-opus.ub.ruhr-uni-bochum.de/opus4/frontdoor/index/index/docId/6580> 6
45. Marlinspike, M., Perrin, T.: The X3DH key agreement protocol (Nov 2016), <https://signal.org/docs/specifications/x3dh/> 2, 17

46. Maurer, U.M.: Abstract models of computation in cryptography (invited paper). In: Smart, N.P. (ed.) 10th IMA International Conference on Cryptography and Coding. LNCS, vol. 3796, pp. 1–12. Springer, Berlin, Heidelberg (Dec 2005) [6](#), [9](#)
47. Menezes, A.: Another look at HMQV. Cryptology ePrint Archive, Report 2005/205 (2005), <https://eprint.iacr.org/2005/205> [15](#)
48. National Institute of Standards and Technology: Digital signature standard (dss). Tech. Rep. FIPS 186-5, U.S. Department of Commerce (2023), <https://www.nist.gov/publications/digital-signature-standard-dss-3> [11](#)
49. Okamoto, T., Pointcheval, D.: The gap-problems: A new class of problems for the security of cryptographic schemes. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Berlin, Heidelberg (Feb 2001) [7](#), [9](#)
50. Pan, J., Riepel, D., Zeng, R.: Key exchange with tight (full) forward secrecy via key confirmation. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part VII. LNCS, vol. 14657, pp. 59–89. Springer, Cham (May 2024) [3](#), [14](#), [33](#)
51. Pan, J., Wagner, B., Zeng, R.: Lattice-based authenticated key exchange with tight security. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part V. LNCS, vol. 14085, pp. 616–647. Springer, Cham (Aug 2023) [3](#), [33](#)
52. Paterson, K.G., Schuldt, J.C.N., Stam, M., Thomson, S.: On the joint security of encryption and signature, revisited. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 161–178. Springer, Berlin, Heidelberg (Dec 2011) [4](#)
53. Patton, C., Shrimpton, T.: Security in the presence of key reuse: Context-separable interfaces and their applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 738–768. Springer, Cham (Aug 2019) [4](#)
54. Perrin, T.: The XEdDSA and VEdDSA signature schemes (Oct 2016), <https://signal.org/docs/specifications/xeddsa/> [4](#), [25](#)
55. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (Jun 2000) [7](#)
56. Qin, X., Cai, C., Yuen, T.H.: One-more unforgeability of blind ECDSA. In: Bertino, E., Shulman, H., Waidner, M. (eds.) ESORICS 2021, Part II. LNCS, vol. 12973, pp. 313–331. Springer, Cham (Oct 2021) [6](#), [9](#), [28](#)
57. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018), <https://www.rfc-editor.org/info/rfc8446> [21](#)
58. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT’97. LNCS, vol. 1233, pp. 256–266. Springer, Berlin, Heidelberg (May 1997) [6](#), [9](#)
59. Signal: Technical information, <https://signal.org/docs/> [2](#)
60. Vatandas, N., Gennaro, R., Ithurburn, B., Krawczyk, H.: On the cryptographic deniability of the Signal protocol. In: Conti, M., Zhou, J., Casalicchio, E., Spognardi, A. (eds.) ACNS 2020, Part II. LNCS, vol. 12147, pp. 188–209. Springer, Cham (Oct 2020) [3](#)
61. Zhandry, M.: To label, or not to label (in generic groups). In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part III. LNCS, vol. 13509, pp. 66–96. Springer, Cham (Aug 2022) [9](#)
62. Zhang, C., Zhou, H.S., Katz, J.: An analysis of the algebraic group model. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part IV. LNCS, vol. 13794, pp. 310–322. Springer, Cham (Dec 2022) [9](#)

## A Security Model of BAKE Protocols

We analyze the BAKE security under a security model combining the prior Bellare-Rogaway(BR)-style model [7] and the Signal’s initial handshake setting [16]. Our model also captures the *multi-test* security with single challenge bit [36,51,50], namely, the adversary is allowed to query the TEST oracle multiple times and all TEST queries use the same challenge bit, which is closer to practical settings when combining BAKE protocols with other protocols (e.g., double ratchet algorithms).

**MULTI-USER, MULTI-SESSION SETTING.** Our model involves multiple parties and multiple sessions between them. We use  $\pi_i^P$  to denote the  $i$ -th session owned by party  $P$ . We also use two global public lists **ipk** and **sspk** to record all identity public keys and semi-static keys. Since a party has multiple semi-static key pairs, for simplicity, we use  $\text{ssid} \in \mathbb{N}$  to identify a semi-static key of a party. More specifically,

- $\Pi[P, i]$  records whether  $\pi_i^P$  has been used.
- **ipk**[ $P$ ] stores the identity public key of party  $P$  (i.e.,  $\text{ipk}_P$ ). We assume that long-term identity public keys are authenticated via some trusted channel, e.g., an out-of-band channel (Signal’s safety numbers). Our analysis focuses on the security under this assumption.
- **sspk**[ $P, \text{ssid}$ ] stores the  $\text{ssid}$ -th semi-static public key of party  $P$ .
- We also use two internal lists **isk** and **sssk** to store all identity and semi-static secret keys, respectively.

In this model, the adversary cannot directly access any  $\pi_i^P$ . Nevertheless, it can remember the indexes  $i$  and  $P$  and use them to reveal some secret information of  $\pi_i^P$  via oracle queries described below.

**ADVERSARY’S CAPABILITIES.** All protocol messages in the model are delivered by the adversary, and it can also replay and tamper with any protocol message. This ability is captured by the SEND and DER oracles (detailed in the security game in Figure 14). Furthermore, the adversary is allowed to leak some secret information of any party and session via the following oracles:

- **CORLT**( $P$ ) returns the identity secret key of  $P$ . We use a Boolean flag list **corLT**[ $P$ ] to indicate whether the party  $P$  has been corrupted via **CORLT**. For an initiator session  $\pi$ , we also use **pcorLT**( $\pi$ ) to record that if the peer of  $\pi$  (i.e.,  $\pi.\text{pid}$ ) has been corrupted before  $\pi$  is completed. This information is used for capturing some forward secrecy properties of Signal’s X3DH protocol.
- **CORSS**( $P, \text{ssid}$ ) returns party  $P$ ’s  $\text{ssid}$ -th semi-static secret key. The Boolean flag **corSS**[ $P, \text{ssid}$ ] records whether **CORSS**( $P, \text{ssid}$ ) has been queried. For simplifying the model, we may also use notation **corSS**[ $\text{sspk}$ ], which is defined as **true** only if  $\text{sspk}$  is some party’s semi-static public key and the corresponding secret key has been corrupted.
- **REVRAND**( $P, i$ ) reveals the randomness used by  $\pi_i^P$  to the adversary and raises the Boolean flag **revRand**[ $\pi_i^P$ ].

- $\text{REVSK}(\mathbf{P}, i)$ : If  $\pi_i^{\mathbf{P}}.\text{SK} \neq \perp$ , then reveal  $\pi_i^{\mathbf{P}}.\text{SK}$  to the adversary and raise the Boolean flag  $\text{revSK}[\pi_i^{\mathbf{P}}]$ .
- $\text{TEST}(\mathbf{P}, i)$ : This oracle is used for defining key indistinguishability. It returns  $\perp$  if  $\pi_i^{\mathbf{P}}.\text{SK} = \perp$ . If the challenge bit  $\beta = 0$ , then return  $\pi_i^{\mathbf{P}}.\text{SK}$ ; Otherwise, an independently uniform key will be returned. All tested sessions are recorded in a set  $\mathcal{S}_{\text{test}}$ . We use the list  $\text{testSK}$  to store the responses of previous TEST queries, so repeated queries to this oracle return the same response.

**SECURITY GAME.** We formalize the BAKE security through the game presented at Figure 14. The game starts with picking a random challenge bit  $\beta$  and generating identity keys and semi-static keys for all parties. The game runs an adversary  $\mathcal{A}$ , which takes all identity and semi-static public keys as its input. During the game,  $\mathcal{A}$  can query all oracles described above. Finally,  $\mathcal{A}$  terminates with a guess  $\beta^*$ , and the game decides whether the adversary wins or loses depending on specific security notion being considered. In this paper, we consider two security goals: *Matching Soundness* and *Key Indistinguishability* (cf. Definition 13).

**MATCHING SOUNDNESS.** Our model captures *matching soundness* through the  $\text{Sound}()$  procedure in Figure 14. This procedure requires that any two distinct sessions sharing a session identifier  $\text{sid} \neq \perp$  derive the same session key and have the same type. If either session is an initiator, the other must be a responder. Consequently, distinct initiator sessions must have distinct session identifiers whenever those identifiers are not  $\perp$ . Furthermore, if three distinct sessions share a session identifier, all three must be in the reduced mode. Thus, in the full mode, at most two sessions may share a session identifier. In the reduced mode, two responder sessions may share a session identifier when the adversary replays the same initiator message to both.

**FRESHNESS AND KEY INDISTINGUISHABILITY.** We use key indistinguishability (IND) to define the security of BAKE session keys, i.e., no PPT adversaries can distinguish whether a session key comes from a real execution or is uniformly random. Since we allow the adversary to access oracles to reveal some secret information of any session, we need some restricted conditions to avoid trivial wins, which is captured by the *Freshness* and *Validity* definitions. The former one requires that if a session  $\pi$ 's key has been tested, then it cannot be revealed and its partner session's key also must not be revealed or tested. The latter one requires that, as long as the adversary does not have sufficient secret information (which will be defined later), a session key should still remain secure. For simplicity, in this paper, we unify these definitions and just call them *freshness*.

In our model, we follow the approach of [23] to define freshness. Namely, we use the  $\text{clean}_{\text{reduced}}$  and  $\text{clean}_{\text{full}}$  predicates to define freshness in different modes. These predicates restrict the amount of secret information available to the adversary, thereby excluding trivially broken test sessions. The definitions of freshness and key indistinguishability are given in Definitions 12 and 13, respectively. The pseudocode for checking freshness and clean predicates are given in Figure 14 and Figure 15, respectively.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Game BAKE<sub>Π</sub><sup>A</sup></b></p> <pre> 01 <math>\beta \xleftarrow{\\$} \{0, 1\}</math> 02 <b>for</b> <math>P \in [N_{LT}]</math> 03   <math>(isk_P, ipk_P) \leftarrow \text{KGenLT}</math> 04   <math>ipk[P] := ipk_P, isk[P] := isk_P</math> 05   <b>for</b> <math>ssid \in [n_{SS}]</math> 06     <math>(sssk_P, sspk_P) \leftarrow \text{KGenSS}(isk_P, ipk_P)</math> 07     <math>sspk[P, ssid] := sspk_P</math> 08     <math>sssk[P, ssid] := sssk_P</math> 09 <math>\beta^* \leftarrow \mathcal{A}^O(ipk, sspk)</math> 10 <b>if</b> Sound() <b>then</b> <b>false</b> 11 <b>return</b> 1 // Break the matching soundness 12 <b>if</b> <math>\exists \pi \in \mathcal{S}_{\text{test}}</math> s.t. Fresh(<math>\pi</math>) = <b>false</b> 13 <b>return</b> 0 // Invalid attack 14 <b>return</b> <math>\llbracket \beta = \beta^* \rrbracket</math> </pre> <p><b>Oracle SEND</b>(<math>P, i, ssid, m, \text{role}, \text{type}</math>)</p> <p>// This oracle is for initializing sessions</p> <pre> 15 <math>\pi_i^P := \Pi[P, i]</math> 16 <b>if</b> <math>\pi_i^P \neq \perp</math>: <b>return</b> <math>\perp</math> // No re-initialization 17 <b>if</b> <math>\text{role} \notin \{I, R\} \vee \text{type} \notin \{\text{full}, \text{reduced}\}</math> 18 <b>return</b> <math>\perp</math> // Invalid query 19 <math>\pi_i^P \leftarrow \text{NewSession}(P, \text{role}, \text{type})</math> 20 <math>(isk, ipk) := (isk[P], ipk[P])</math> 21 <math>(sssk, sspk) := (sssk[P, ssid], sspk[P, ssid])</math> 22 <b>Case</b> <math>\text{role} = R</math>: // Responder creates pre-key bundles 23   // <math>m</math> will be ignored in this case 24   <math>(\pi_i^P, m') := \text{InitR}(\pi_i^P, isk, ipk, sssk, sspk)</math> 25 <b>Case</b> <math>\text{role} = I</math>: // Initiator session creation 26   // <math>ssid</math> will be ignored in this case 27   <math>(\pi_i^P, m') := \text{InitI}(\pi_i^P, isk, ipk, m)</math> 28   <math>\text{pcorLT}[\pi_i^P] := \text{corLT}[\pi_i^P.\text{pid}]</math> 29 <math>\Pi[P, i] := \pi_i^P</math> // The session has been initialized 30 <b>return</b> <math>m'</math> </pre> <p><b>Oracle CORLT</b>(<math>P</math>)</p> <pre> 31 <math>\text{corLT}[P] := \text{true}</math> 32 <b>return</b> <math>isk[P]</math> </pre> <p><b>Oracle CORSS</b>(<math>P, ssid</math>)</p> <pre> 33 <math>sspk := sspk[P, ssid]</math> 34 <math>\text{corSS}[sspk] := \text{true}</math> 35 <b>return</b> <math>sssk[P, ssid]</math> </pre> | <p><b>Procedure Sound</b>() // Matching soundness</p> <pre> 36 <b>return</b> <math>\forall</math> distinct sessions <math>\pi, \pi', \pi''</math> ( 37   <math>(\pi.\text{sid} = \pi'.\text{sid} \neq \perp \implies</math> 38     <math>\left( \begin{aligned} &amp;\pi.SK = \pi'.SK \wedge \pi.\text{type} = \pi'.\text{type} \\ &amp;\wedge (\pi.\text{role} = I \implies \pi'.\text{role} = R) \end{aligned} \right)</math> 39   ) 40   <math>\wedge \left( \begin{aligned} &amp;\pi.\text{sid} = \pi'.\text{sid} = \pi''.\text{sid} \\ &amp;\implies \pi.\text{type} = \text{reduced} \end{aligned} \right)</math> 41 ) </pre> <p><b>Procedure Fresh</b>(<math>\pi</math>) // Freshness &amp; Validity</p> <pre> 39 <b>if</b> revSK[<math>\pi</math>]: <b>return</b> <b>false</b> 40 <b>if</b> <math>\exists \pi' \neq \pi</math> s.t. <math>\pi'.\text{sid} = \pi.\text{sid}</math> 41   <math>\wedge (\text{revSK}[\pi'] \vee \pi' \in \mathcal{S}_{\text{test}})</math>: 42 <b>return</b> <b>false</b> 43 <b>if</b> <math>\text{clean}_{\pi.\text{type}}^\Pi(\pi) = \text{false}</math>: <b>return</b> <b>false</b> 44 <b>return</b> <b>true</b> </pre> <p><b>Oracle DER</b>(<math>P, i, ssid, m</math>)</p> <p>// This oracle is for completing a responder session</p> <pre> 45 <math>\pi_i^P := \Pi[P, i]</math> 46 <b>if</b> <math>\pi_i^P = \perp</math>: <b>return</b> <math>\perp</math> 47 <b>if</b> <math>\pi_i^P.SK \neq \perp</math>: <b>return</b> <math>\perp</math> // No session reuse 48 <math>(isk, ipk) := (isk[P], ipk[P])</math> 49 <math>(sspk, sssk) := (sspk[P, ssid], sssk[P, ssid])</math> 50 <math>\pi_i^P := \text{DerR}(\pi_i^P, isk, ipk, sssk, sspk, m)</math> 51 <math>\Pi[P, i] := \pi_i^P</math> 52 <b>return</b> <math>\varepsilon</math> </pre> <p><b>Oracle TEST</b>(<math>P, i</math>)</p> <pre> 53 <b>if</b> <math>\pi_i^P.SK = \perp</math>: <b>return</b> <math>\perp</math> 54 // This session has not accepted a key 55 <b>if</b> testSK[<math>\pi_i^P</math>] = <math>\perp</math> // First test query of <math>\pi_i^P</math> 56   <math>K_0 := \pi_i^P.SK, K_1 \xleftarrow{\\$} \text{BAKE.SK}</math> 57   testSK[<math>\pi_i^P</math>] := <math>K_\beta</math> // Real-or-Random 58   <math>\mathcal{S}_{\text{test}} := \mathcal{S}_{\text{test}} \cup \{\pi_i^P\}</math> 59 <b>return</b> testSK[<math>\pi_i^P</math>] </pre> <p><b>Oracle REV RAND</b>(<math>P, i</math>)</p> <pre> 59 <b>if</b> <math>\pi_i^P.\text{rand} \neq \perp</math> 60   revRand[<math>\pi_i^P</math>] := <b>true</b> 61 <b>return</b> <math>\pi_i^P.\text{rand}</math> </pre> <p><b>Oracle REV SK</b>(<math>P, i</math>)</p> <pre> 62 <b>if</b> <math>\pi_i^P.SK \neq \perp</math> 63   revSK[<math>\pi_i^P</math>] := <b>true</b> 64 <b>return</b> <math>\pi_i^P.SK</math> </pre> |
| <p><b>Procedure clean<sub>reduced</sub><sup>X3DH</sup>(<math>\pi</math>), clean<sub>reduced</sub><sup>XHMQV</sup>(<math>\pi</math>)</b></p> <pre> 65 flag := <b>false</b> 66 flag := (Eph-SS(<math>\pi</math>) or Eph-LT(<math>\pi</math>) 67   or LT-SS(<math>\pi</math>) or LT-LT(<math>\pi</math>)) 67 <b>return</b> flag </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <p><b>Procedure clean<sub>full</sub><sup>X3DH</sup>(<math>\pi</math>), clean<sub>full</sub><sup>XHMQV</sup>(<math>\pi</math>)</b></p> <pre> 68 flag := <b>false</b> 69 flag := (Eph-SS(<math>\pi</math>) or Eph-LT(<math>\pi</math>) or Eph-Eph(<math>\pi</math>) 70   or (LT-SS(<math>\pi</math>) or LT-LT(<math>\pi</math>) or LT-Eph(<math>\pi</math>)) 70 <b>return</b> flag </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

**Fig. 14.** The security game for defining BAKE security. The adversary has access to oracles  $O := \{\text{SEND}, \text{DER}, \text{CORLT}, \text{CORSS}, \text{REVRAND}, \text{REVSK}, \text{TEST}\}$ . The helper procedures for checking freshness are presented in Figure 15. The highlighted parts only exist in the security game for XHMQV.

**Definition 12 (Freshness and Validity).** We say a session  $\pi$  is fresh if the following two conditions hold:

1.  $\pi$ 's session key is unrevealed, i.e.,  $\text{revSK}[\pi] = \text{false}$ .
2. If there exists a session  $\pi' \neq \pi$  such that  $\pi'$  is partnered with  $\pi$  (i.e.,  $\pi.\text{sid} = \pi'.\text{sid}$ ), then  $\pi'$  must not have been tested or revealed.

Furthermore, if  $\pi$  had been tested (i.e.,  $\pi \in \mathcal{S}_{\text{test}}$ ), then we say the attack on  $\pi$  is valid if the clean predicate  $\text{clean}_{\pi.\text{type}}^{\Pi}(\pi)$  outputs **true**.

We consider two specific clean predicates:  $\text{clean}_{\pi.\text{type}}^{\text{X3DH}}$  and  $\text{clean}_{\pi.\text{type}}^{\text{XMQV}}$ . The former one defines the security of Signal's X3DH protocol, which captures the (LT, SS), (E, LT), (E, SS), and (E,E) (in the full mode) leakage scenarios described below, while the latter one defines the security of XMQV, which captures all leakage scenarios described below:

1. (LT,LT): Both the owner and peer of  $\pi$  are uncorrupted (cf. LT-LT in Line 81).
2. (LT,SS): The initiator of  $\pi$  is uncorrupted, and the semi-static key used in the session is secure, namely, it is uncorrupted and either (1) the semi-static public key is valid in the initiator's view, or (2) the responder was uncorrupted at the time the session obtained the pre-key bundle. (cf. Line 28 and LT-SS in Line 78).
3. (LT,E): The initiator of  $\pi$  is uncorrupted and the randomness used in the responder session  $\pi'$  related to  $\pi$  (either  $\pi = \pi'$  or  $\pi'$  is partnered with  $\pi$ ) is unrevealed. If  $\pi'$  does not exist, then  $\pi$  is considered not to satisfy this condition (cf. LT-Eph in Line 82).
4. (E,LT): The randomness used in the initiator session  $\pi'$  related to  $\pi$  is unrevealed and the responder of  $\pi$  is uncorrupted. If  $\pi'$  does not exist, then  $\pi$  is considered not to satisfy this condition (cf. Eph-LT in Line 79).
5. (E,SS): The randomness used in the initiator session  $\pi'$  related to  $\pi$  is unrevealed and the semi-static key used in  $\pi$  is secure (cf. Line 28 and Eph-SS in Line 80).
6. (E,E): The randomness used in  $\pi$  and its partnered session are unrevealed.  $\pi$  is considered not to satisfy this condition if it does not have any partnered session (cf. Eph-Eph in Line 77).

The procedure  $\text{clean}_{\pi.\text{type}}^{\Pi}$  outputs **true** as long as at least one of the above conditions defined for  $\Pi$  and  $\pi.\text{type}$  is satisfied.

An adversary  $\mathcal{A}$  breaks a BAKE protocol  $\Pi$  if (1)  $\mathcal{A}$  breaks the matching soundness of  $\Pi$  in  $\text{BAKE}_{\Pi}^A$ , or (2)  $\mathcal{A}$  successfully distinguishes a session key of a fresh session from uniformly random keys. The formal BAKE security definition is given in Definition 13, and the pseudocode for checking the winning conditions is in Lines 10 to 14.

**Definition 13 (BAKE Security).** Let  $\Pi$  be a BAKE protocol. Let the numbers of parties, semi-static keys of each party, and semi-static keys be  $N_{\text{LT}}$ ,  $n_{\text{SS}}$ , and  $N_{\text{SS}} (:= N_{\text{LT}} \cdot n_{\text{SS}})$ , respectively. For any adversary  $\mathcal{A}$  against  $\Pi$  in  $\text{BAKE}_{\Pi}^A$ , making at most  $Q_{\text{LT}}$  queries to CORLT,  $Q_{\text{SS}}$  queries to CORSS,  $N_{\text{ses}}$  queries to SEND

|                                                                                                                                                                                                    |                                                                                                                                                                                                                                           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>peerEph(<math>\pi</math>)</u>                                                                                                                                                                   | <u>LT-SS(<math>\pi</math>)</u>                                                                                                                                                                                                            |
| 71 <b>Case</b> $\pi.\text{role} = \text{I}$ :                                                                                                                                                      | 78 <b>return</b> $(\pi.\text{role} = \text{I} \wedge \neg\text{corLT}[\pi.\text{oid}] \wedge \text{peerSS}(\pi))$<br>$\vee (\pi.\text{role} = \text{R} \wedge \neg\text{corLT}[\pi.\text{pid}] \wedge \neg\text{corSS}[\pi.\text{sspk}])$ |
| 72 <b>return</b> $\exists \pi' : \left( \begin{array}{l} \pi'.\text{role} = \text{R} \\ \wedge \pi.\text{cid} = \pi'.\text{cid} \\ \wedge \neg\text{revRand}(\pi') \end{array} \right)$            | <u>Eph-LT(<math>\pi</math>)</u>                                                                                                                                                                                                           |
| 73 <b>Case</b> $\pi.\text{role} = \text{R}$ :                                                                                                                                                      | 79 <b>return</b> $(\pi.\text{role} = \text{R} \wedge \neg\text{corLT}[\pi.\text{oid}] \wedge \text{peerEph}(\pi))$<br>$\vee (\pi.\text{role} = \text{I} \wedge \neg\text{corLT}[\pi.\text{pid}] \wedge \neg\text{revRand}(\pi))$          |
| 74 <b>return</b> $\exists \pi' : \left( \begin{array}{l} \pi'.\text{role} = \text{I} \\ \wedge \pi.\text{sid} = \pi'.\text{sid} \\ \wedge \neg\text{revRand}(\pi') \end{array} \right)$            | <u>Eph-SS(<math>\pi</math>)</u>                                                                                                                                                                                                           |
| 75 <b>return false</b>                                                                                                                                                                             | 80 <b>return</b> $(\pi.\text{role} = \text{I} \wedge \neg\text{revRand}(\pi) \wedge \text{peerSS}(\pi))$<br>$\vee (\pi.\text{role} = \text{R} \wedge \neg\text{corSS}[\pi.\text{sspk}] \wedge \text{peerEph}(\pi))$                       |
| <u>peerSS(<math>\pi</math>)</u>                                                                                                                                                                    | <u>LT-LT(<math>\pi</math>)</u>                                                                                                                                                                                                            |
| 76 <b>return</b> $\left( \begin{array}{l} \pi.\text{sspk} \in \text{sspk}[\pi.\text{pid}, \cdot] \\ \vee \neg\text{pcorLT}[\pi] \end{array} \right)$<br>$\wedge \neg\text{corSS}[\pi.\text{sspk}]$ | 81 <b>return</b> $\left( \begin{array}{l} \neg\text{corLT}[\pi.\text{oid}] \\ \wedge \neg\text{corLT}[\pi.\text{pid}] \end{array} \right)$                                                                                                |
| <u>Eph-Eph(<math>\pi</math>)</u>                                                                                                                                                                   | <u>LT-Eph(<math>\pi</math>)</u>                                                                                                                                                                                                           |
| 77 <b>return</b> $\neg\text{revRand}(\pi) \wedge \text{peerEph}(\pi)$                                                                                                                              | 82 <b>return</b> $(\pi.\text{role} = \text{I} \wedge \neg\text{corLT}[\pi.\text{oid}] \wedge \text{peerEph}(\pi))$<br>$\vee (\pi.\text{role} = \text{R} \wedge \neg\text{corLT}[\pi.\text{pid}] \wedge \neg\text{revRand}(\pi))$          |

**Fig. 15.** The helper predicates for determining freshness in  $\text{clean}_{\pi.\text{type}}^{\Pi}(\pi)$  (cf. Figure 14).

and DER,  $Q_{\text{Eph}}$  queries to REV RAND,  $Q_{\text{test}}$  queries to TEST, and  $Q_{\text{revSK}}$  queries to REV SK, we define the advantage of  $\mathcal{A}$  as

$$\text{Adv}_{\Pi}^{\text{BAKE}}(\mathcal{A}) := \left| \Pr \left[ \text{BAKE}_{\Pi}^{\mathcal{A}} = 1 \right] - \frac{1}{2} \right|,$$

*Remark 3 (Explanations on the clean predicates).* When defining  $\text{clean}_{\text{reduced}}$  and  $\text{clean}_{\text{full}}$ , we use six predicates  $\text{XX}_1\text{-XX}_2$  (cf. Figure 15) to classify different attack cases based on the information that the adversary *does not* have, where  $\text{XX}_1, \text{XX}_2 \in \{\text{LT}, \text{SS}, \text{Eph}\}$ . Here LT, SS, and Eph correspond to the identity (i.e., long-term) key information, semi-static key information, and ephemeral key (i.e., one-time pre-key) information, respectively. X3DH covers four cases: LT-SS, Eph-Eph, Eph-SS, and Eph-LT, while XHMQV covers all six cases.

More specifically,  $\text{LT-XX}_2(\pi)$  and  $\text{XX}_1\text{-LT}$  capture the security corresponding to authentication of the initiator and responder, respectively. Intuitively, this means that the session key should still remain secure if the long-term key, which binds the party's identity, is secure. When  $\text{XX}_2 \neq \text{LT}$  (or  $\text{XX}_1 \neq \text{LT}$ ), these predicates additionally capture key-compromise impersonation (KCI) security, as the peer party may be corrupted. The Eph-Eph predicate represents weak forward secrecy (wFS). We refer readers to [23,24] for more discussions on how these predicates capture security.

## B Security Proof of X3DH<sup>+</sup> (Theorem 1)

**Game  $\mathbf{G}_0$ :** This game is the  $\text{BAKE}_{\Pi}^{\mathcal{A}}$  game instantiated with X3DH<sup>+</sup>, where the two KDF functions for deriving session keys are simulated via random oracles. Since our proof is in the ROM, this change is conceptual and we have

$$\Pr \left[ \text{BAKE}_{\text{X3DH}^+}^{\mathcal{A}}(\mathcal{A}) = 1 \right] = \Pr \left[ \mathbf{G}_0^{\mathcal{A}} = 1 \right]$$

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Game <math>G_0</math>-<math>G_6</math></b><br>01 $\beta \xleftarrow{\$} \{0, 1\}$<br>02 <b>for</b> $P \in [N_{LT}]$<br>03 $(isk[P], ipk[P]) := (u, U) \leftarrow \text{KGenLT}$<br>04 <b>for</b> $ssid \in [n_{SS}]$<br>05 $(s, (S, \sigma)) \leftarrow \text{KGenSS}(u, U)$<br>06 $(\sigma, q, h) \xleftarrow{\$} \text{SIMsign}(U, S)$ // $G_5$ - $G_6$<br>07 $\mathcal{L}_{RO} := \mathcal{L}_{RO} \cup \{(q, h)\}$ // $G_5$ - $G_6$<br>08 $(sssk[P, ssid], sspk[P, ssid]) := (s, (S, \sigma))$<br>09 $\beta^* \leftarrow \mathcal{A}^O(ipk, sspk)$<br>10 <b>if</b> $\text{Sound}() = \text{false}$ : <b>return</b> 1<br>11 <b>if</b> $\exists \pi \in S_{\text{test}}$ s.t. $\text{Fresh}(\pi) = \text{false}$ : <b>return</b> 0<br>12 <b>return</b> $\llbracket \beta = \beta^* \rrbracket$                                                                                                                                | <b>Oracle <math>\text{SEND}(P, i, ssid, m, \text{role}, \text{type})</math></b><br>44 <b>if</b> $\pi := \Pi[P, i] \neq \perp$ : <b>return</b> $\perp$<br>45 <b>if</b> $\text{role} \notin \{I, R\}$<br>46 $\vee \text{type} \notin \{\text{full}, \text{reduced}\}$<br>47 <b>return</b> $\perp$<br>48 $r \xleftarrow{\$} \mathbb{Z}_p^*$<br>49 $\pi.(\text{oid}, \text{role}, \text{rand}) := (P, \text{role}, r)$<br>50 <b>Case</b> $\text{role} = R$ // $\text{Init}_R$<br>51 $B := P, (b, B) := (isk[P], ipk[P])$<br>52 $s := sssk[P, ssid]$<br>53 $(S, \sigma) := sspk[P, ssid]$<br>54 <b>if</b> $\text{type} = \text{full}$ : $y := r, Y := g^y$<br>55 <b>else</b> $Y := \perp$<br>56 $m' := m_{\text{bundle}} := (B, (S, \sigma), Y)$<br>57 $\pi.(\text{pid}, \text{type}) := (*, \text{type})$<br>58 $\pi.\text{state} := (y, Y)$<br>59 $\pi.\text{cid} := (B, (S, \sigma), Y)$<br>60 <b>Case</b> $\text{role} = I$ : // $\text{Init}_I$<br>61 $(B, (S, \sigma), Y) := m_{\text{bundle}}$<br>62 $(a, A) := (isk[P], ipk[P])$ , $B := ipk[B]$<br>63 <b>if</b> $\text{Verify}(B, \sigma, S) \neq \text{true}$ : <b>return</b> $\perp$<br>64 <b>else</b> $\pi.sspk := (S, \sigma)$<br>65 <b>if</b> $\neg \text{corLT}[B]$ // $G_4$ - $G_6$<br>66 $\wedge (S, \sigma) \notin sspk[B, \cdot]$ // $G_4$ - $G_6$<br>67 <b>abort and output</b> $\beta$ // $G_4$ - $G_6$<br>68 <b>if</b> $Y \neq \perp$ : $\pi.\text{type} := \text{full}$<br>69 <b>else</b> $\pi.\text{type} := \text{reduced}$<br>70 $x := \pi.\text{rand}, X := g^x$<br>71 $\text{ctxt}_r := (A, X, B, S, \sigma)$<br>72 $\text{ctxt}_f := (A, X, B, S, Y, \sigma)$<br>73 $\text{DH} := (S^a, B^x, S^x)$<br>74 $m' := (A, X)$<br>75 <b>if</b> $\pi.\text{type} = \text{full}$<br>76 $\text{DH} := \text{DH} \parallel Y^x$<br>77 $K := \text{KDF}_f(\text{DH}, \text{ctxt}_f)$<br>78 <b>else</b> $K := \text{KDF}_r(\text{DH}, \text{ctxt}_r)$<br>79 $K := \text{PatchSK}(A, X, B, S, Y, \sigma)$ // $G_3$ - $G_6$<br>80 $\pi.(\text{pid}, \text{SK}) := (B, K)$<br>81 $\pi.\text{cid} := (B, (S, \sigma), Y)$<br>82 $\pi.\text{sid} := (A, X, B, S, Y, \sigma)$<br>83 $\text{pcorLT}[\pi_i^P] := \text{corLT}[\pi_i^P.\text{pid}]$<br>84 $\Pi[P, i] := \pi_i^P$<br>85 <b>return</b> $m'$ |
| <b>Oracle <math>\text{DER}(P, i, ssid, m)</math></b><br>13 $\pi_i^P := \Pi[P, i]$<br>14 <b>if</b> $\pi_i^P = \perp \vee \pi_i^P.\text{SK} \neq \perp$ : <b>return</b> $\perp$<br>15 $B := P, (b, B) := (isk[P], ipk[P])$<br>16 $s := sssk[P, ssid]$ , $(S, \sigma) := sspk[P, ssid]$<br>17 $(y, Y) := \pi.\text{state}$<br>18 $(A, X) := m, A := ipk[A]$<br>19 $\text{ctxt}_r := (A, X, B, S, \sigma)$<br>20 $\text{ctxt}_f := (A, X, B, S, Y, \sigma)$<br>21 $\text{DH} := (A^s, X^b, X^s)$<br>22 <b>if</b> $\pi.\text{type} = \text{full}$<br>23 $\text{DH} := \text{DH} \parallel X^y, K := \text{KDF}_f(\text{DH}, \text{ctxt}_f)$<br>24 <b>else</b> $K := \text{KDF}_r(\text{DH}, \text{ctxt}_r)$<br>25 $K := \text{PatchSK}(A, X, B, S, Y, \sigma)$ // $G_3$ - $G_6$<br>26 $\pi.(\text{pid}, \text{SK}) := (A, K)$<br>27 $\pi.\text{sid} := (A, X, B, S, Y, \sigma)$<br>28 $\Pi[P, i] := \pi_i^P$ ; <b>return</b> $\epsilon$ |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Oracle <math>\text{KDF}_f(Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma)</math></b><br>29 <b>if</b> $Z_1 = \text{cdh}(A, S)$ // $G_2$ - $G_6$<br>30 $\wedge Z_2 = \text{cdh}(X, B)$ // $G_2$ - $G_6$<br>31 $\wedge Z_3 = \text{cdh}(X, S)$ // $G_2$ - $G_6$<br>32 $\wedge Z_4 = \text{cdh}(X, Y)$ // $G_2$ - $G_6$<br>33 <b>if</b> $\text{cleanDH}(A, X, B, S, Y) \neq \perp$ // $G_6$<br>34 <b>abort and output</b> $\beta$ // $G_6$<br>35 <b>return</b> $\text{PatchSK}(A, X, B, S, Y, \sigma)$ // $G_2$ - $G_6$<br>36 <b>if</b> $\mathcal{L}_{\text{KDF}_f}[Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma] = \perp$<br>37 $K \xleftarrow{\$} \text{SK}$<br>38 $\mathcal{L}_{\text{KDF}_f}[Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma] := K$<br>39 <b>return</b> $\mathcal{L}_{\text{KDF}_f}[Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma]$                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Oracle <math>\text{RO}(q)</math></b><br>40 <b>if</b> $\mathcal{L}_{\text{RO}}[q] = \perp$<br>41 $h \xleftarrow{\$} \text{RO.image}$<br>42 $\mathcal{L}_{\text{RO}}[q] := h$<br>43 <b>return</b> $\mathcal{L}_{\text{RO}}[q]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <b>Procedure <math>\text{PatchSK}(A, X, B, S, Y, \sigma)</math></b><br>86 <b>if</b> $\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] = \perp$<br>87 $K \xleftarrow{\$} \text{SK}$<br>88 $\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] := K$<br>89 <b>return</b> $\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

**Fig. 16.** Games  $G_0$ - $G_6$  in proving Theorem 1. The oracles  $\text{CORLT}$ ,  $\text{CORSS}$ ,  $\text{REVRAND}$ ,  $\text{REVSK}$  and  $\text{TEST}$  and auxiliary procedures  $\text{Fresh}$  and  $\text{Sound}$  are the same as in Figure 14. The oracle  $\text{KDF}_f$  (which is similar to  $\text{KDF}_r$ ) and auxiliary procedure  $\text{cleanDH}$  (for  $G_6$ ) are defined in Figure 17.

| <b>Procedure</b> $\text{cleanDH}(A, X, B, S, Y)$                                                           | <b>Oracle</b> $\text{KDF}_r(Z_1, Z_2, Z_3, A, X, B, S, \sigma)$                                    |
|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| 01 Find $\pi$ such that $\pi.\text{sid} = (A, X, B, S, Y, \cdot)$                                          | 20 <b>if</b> $Z_1 = \text{cdh}(A, S)$ // $\mathbf{G}_2\text{-}\mathbf{G}_5, \mathbf{G}_6$          |
| 02 <b>if</b> $\pi = \perp$                                                                                 | 21 $\wedge Z_2 = \text{cdh}(X, B)$ // $\mathbf{G}_2\text{-}\mathbf{G}_5, \mathbf{G}_6$             |
| 03 <b>return</b> $\perp$ // No corresponding sessions                                                      | 22 $\wedge Z_3 = \text{cdh}(X, S)$ // $\mathbf{G}_2\text{-}\mathbf{G}_5, \mathbf{G}_6$             |
| 04 Find $A$ s.t. $\text{ipk}[A] = A$                                                                       | 23 <b>return</b> $\text{PatchSK}(A, X, B, S, \perp, \sigma)$ // $\mathbf{G}_2\text{-}\mathbf{G}_5$ |
| 05 Find $B$ s.t. $\text{ipk}[B] = B$                                                                       | 24 <b>if</b> $\text{cleanDH}(A, X, B, S, \perp) \neq \perp$ // $\mathbf{G}_6$                      |
| 06 Find $\text{ssid}$ s.t. $\text{sspk}[B, \text{ssid}] = (S, \cdot)$                                      | 25 <b>abort</b> and <b>output</b> $\beta$ // $\mathbf{G}_6$                                        |
| 07 Find $i$ s.t. $x = \Pi[A, i].\text{rand}$ and $X = g^x$                                                 | 26 <b>if</b> $\mathcal{L}_{\text{KDF}_r}[Z_1, Z_2, Z_3, A, X, B, S, \sigma] = \perp$               |
| 08 Find $j$ s.t. $y = \Pi[B, j].\text{rand}$ and $Y = g^y$                                                 | 27 $K \xleftarrow{\$} \mathcal{SK}$                                                                |
| 09 <b>if</b> $\text{ssid} \neq \perp \wedge \neg \text{corLT}[A] \wedge \neg \text{corSS}[B, \text{ssid}]$ | 28 $\mathcal{L}_{\text{KDF}_r}[Z_1, Z_2, Z_3, A, X, B, S, \sigma] := K$                            |
| 10 <b>return</b> 1 // Detect a valid LT-SS attack                                                          | 29 <b>return</b> $\mathcal{L}_{\text{KDF}_r}[Z_1, Z_2, Z_3, A, X, B, S, \sigma]$                   |
| 11 <b>if</b> $i \neq \perp \wedge \neg \text{revRand}[A, i] \wedge \neg \text{corLT}[B]$                   |                                                                                                    |
| 12 <b>return</b> 2 // Detect a valid Eph-LT attack                                                         |                                                                                                    |
| 13 <b>if</b> $i \neq \perp \wedge \text{ssid} \neq \perp$                                                  |                                                                                                    |
| 14 $\wedge \neg \text{revRand}[A, i] \wedge \neg \text{corSS}[B, \text{ssid}]$                             |                                                                                                    |
| 15 <b>return</b> 3 // Detect a valid Eph-SS attack                                                         |                                                                                                    |
| 16 <b>if</b> $i \neq \perp \wedge j \neq \perp$                                                            |                                                                                                    |
| 17 $\wedge \neg \text{revRand}[A, i] \wedge \neg \text{revRand}[B, j]$                                     |                                                                                                    |
| 18 <b>return</b> 4 // Detect a valid Eph-Eph attack                                                        |                                                                                                    |
| 19 <b>return</b> $\perp$                                                                                   |                                                                                                    |

**Fig. 17.** The oracle  $\text{KDF}_r$  and auxiliary procedure  $\text{cleanDH}$  in Figure 16.

**Game  $\mathbf{G}_1$ :** In this game, we exclude any collision on the long-term public keys, semi-static public keys, and ephemeral public keys. If such a collision happens, we abort and output a random bit. This game ensures that all public keys are unique, which is a common assumption in the security analysis of AKE protocols. In the statement of Theorem 1, we assume that there are at most  $N_{\text{LT}}$  long-term keys,  $N_{\text{SS}}$  semi-static keys, and  $N_{\text{ses}}$  sessions (i.e., ephemeral keys) generated during the game. The probability of this event can be bounded by a standard birthday bound, and we have

$$|\Pr[\mathbf{G}_0^A = 1] - \Pr[\mathbf{G}_1^A = 1]| \leq \frac{(N_{\text{LT}} + N_{\text{SS}} + N_{\text{ses}})^2}{2p}$$

**Game  $\mathbf{G}_2$ :** We change the simulation of  $\text{KDF}_f$  and  $\text{KDF}_r$ . When  $\mathcal{A}$  queries  $\text{KDF}_f$  or  $\text{KDF}_r$  with the correct DH tuple, then the key returned by the oracle is generated by a special procedure  $\text{PatchSK}$  that we define in Figure 16. For example, when  $\mathcal{A}$  queries  $\text{KDF}_f(Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma)$ , where  $Z_1 = \text{cdh}(A, S)$ ,  $Z_2 = \text{cdh}(X, B)$ ,  $Z_3 = \text{cdh}(X, S)$ , and  $Z_4 = \text{cdh}(X, Y)$ , then the oracle will return  $\text{PatchSK}(A, X, B, S, Y, \sigma)$ . Similar changes are made for  $\text{KDF}_r$ .

In the auxiliary procedure  $\text{PatchSK}$ , we simulate its output in a random oracle manner. Specifically, we use a list  $\mathcal{L}_{\text{SK}}$  to store the session keys generated by the oracle. If the input tuple  $(A, X, B, S, Y, \sigma)$  has been queried before, then we return the same session key as before. Otherwise, we generate a new random session key and store it in the list.

We argue that this change does not affect the probability of  $\mathcal{A}$  winning the game. First,  $\text{PatchSK}$  simulates the output of the KDF functions in a random oracle manner (e.g.,  $\text{RO}(A, X, B, S, Y, \sigma)$ ). Second, by the algebraic structure

of the DH tuples, if we fix  $(A, X, B, S, Y)$ , then its  $\text{X3DH}^+$  DH combinations are uniquely determined because they depend only on the fixed exponents. In other words, there is exactly one correct 4-tuple  $(Z_1, Z_2, Z_3, Z_4)$  that matches the DH computations in  $\text{X3DH}^+$ . Hence,  $\text{KDF}_f(Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma)$  is independent of  $\text{KDF}_f(Z'_1, Z'_2, Z'_3, Z'_4, A, X, B, S, Y, \sigma)$  if  $(Z_1, Z_2, Z_3, Z_4) \neq (Z'_1, Z'_2, Z'_3, Z'_4)$ . Therefore, this change does not affect the distribution of the outputs of the KDF functions, and we have

$$\Pr [\mathbf{G}_1^{\mathcal{A}} = 1] = \Pr [\mathbf{G}_2^{\mathcal{A}} = 1]$$

In  $\mathbf{G}_2$ , although the adversary can query KDFs on some DH public keys that are not generated by the honest parties and thus checking CDH values requires the knowledge of the corresponding secret keys, it does not cause any problem for the security analysis, since all reductions shown later have DDH oracle to check the correctness of the DH tuples in KDF queries.

**Game  $\mathbf{G}_3$ :** In this game, we change the simulation of  $\text{Init}_I$  and  $\text{Der}_R$  such that they directly call the **PatchSK** procedure to compute the session keys. This change is conceptual since the call to  $\text{KDF}_f$  and  $\text{KDF}_r$  from  $\text{Init}_I$  or  $\text{Der}_R$  will eventually call **PatchSK** in  $\mathbf{G}_2$ . Hence, this change does not affect the probability of  $\mathcal{A}$  winning the game, and we have

$$\Pr [\mathbf{G}_2^{\mathcal{A}} = 1] = \Pr [\mathbf{G}_3^{\mathcal{A}} = 1].$$

From now on,  $\text{Init}_I$  and  $\text{Der}_R$  can compute session keys without knowing the secret keys, which is crucial for the security analysis in the following games.

**Game  $\mathbf{G}_4$ :** This game excludes an event **ForgeSS** that  $\mathcal{A}$  successfully forges a valid signature on a semi-static public key of an uncorrupted party. Namely, when  $\mathcal{A}$  sends a pre-key bundle message  $(B, (S, \sigma), Y)$  to  $A$  (via querying  $\text{Init}_I$ ), if the signature  $\sigma$  is valid but the semi-static public key  $S$  in the message is not generated by  $B$  and  $B$  is not corrupted, then we say **ForgeSS** happens. Once **ForgeSS** happens, the game aborts and outputs a random bit.

Before bounding the probability of **ForgeSS**, we first discuss the impact of excluding **ForgeSS**. If **ForgeSS** never happens, then it means that, for any initiator session  $\pi$ , as long as  $\text{pcorLT}[\pi] = \text{false}$  (the responder peer was not corrupted when  $\pi$  was created), we have that the used semi-static DH public key  $S$  in  $\pi$  must be honest, i.e., generated by the responder peer  $\pi.\text{pid}$ . This property is important for the next game.

To bound **ForgeSS**, we construct a reduction from the  $\text{EUF}^c\text{-1CMA-DDH}$  security of the signature scheme  $\Sigma$  in  $\text{X3DH}$ . The reduction  $\mathcal{F}$  is presented in Figure 18.  $\mathcal{F}$  takes as input a list of long-term public keys  $U_1, \dots, U_{N_{\text{LT}}}$ , and has access to the oracles **SIGN**, **CORR**, and **DDH**, as defined in Definition 8. All signatures are simulated via the **SIGN** oracle (without using the secret keys) and if  $\mathcal{A}$  corrupts a party, then  $\mathcal{F}$  uses the **CORR** oracle to answer the corruption query. The **DDH** oracle is used to check if the DH tuples in the  $\text{KDF}_f$  queries are correct. If **ForgeSS** happens, then  $\mathcal{F}$  uses  $(P^*, \sigma^*, S^*)$  to record the forgery and returns it. By the definition of **ForgeSS**, this forgery is a fresh and valid

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Reduction</b> $\mathcal{F}^{\text{SIGN}, \text{CORR}, \text{DDH}}(U_1, \dots, U_{N_{\text{LT}}})$<br>01 $\beta \xleftarrow{\$} \{0, 1\}, (\mathbf{P}^*, \sigma^*, S^*) \xleftarrow{\$} [N_{\text{LT}}] \times \Sigma.S \times \mathbb{G}$<br>02 <b>for</b> $\mathbf{P} \in [N_{\text{LT}}]$<br>03 $(\text{isk}[\mathbf{P}], \text{ipk}[\mathbf{P}]) := (\perp, U_{\mathbf{P}})$<br>04 <b>for</b> $\text{ssid} \in [n_{\text{SS}}]$<br>05 $s \xleftarrow{\$} \mathbb{Z}_p, S := g^s, \sigma \leftarrow \text{SIGN}(\mathbf{P}, S)$<br>06 $(\text{sssk}[\mathbf{P}, \text{ssid}], \text{sspk}[\mathbf{P}, \text{ssid}]) := (s, (S, \sigma))$<br>07 $\beta^* \leftarrow \mathcal{A}^O(\text{ipk}, \text{sspk})$<br>08 <b>return</b> $(\mathbf{P}^*, \sigma^*, S^*)$ // Random forgery<br><br><b>Oracle</b> $\text{KDF}_r(Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma)$<br>09 <b>if</b> $\text{DDH}(A, S, Z_1)$ // $Z_1 = \text{cdh}(A, S)$<br>10 $\wedge \text{DDH}(X, B, Z_2)$ // $Z_2 = \text{cdh}(X, B)$<br>11 $\wedge \text{DDH}(X, S, Z_3)$ // $Z_3 = \text{cdh}(X, S)$<br>12 $\wedge \text{DDH}(X, Y, Z_4)$ // $Z_4 = \text{cdh}(X, Y)$<br>13 <b>return</b> $\text{PatchSK}(A, X, B, S, Y, \sigma)$<br>14 ... // Lazy sampling as in $\mathbf{G}_4$ | <b>Oracle</b> $\text{KDF}_r(Z_1, Z_2, Z_3, A, X, B, S, \sigma)$<br>15 ...similar to $\text{KDF}_r$ ...<br><br><b>Oracle</b> $\text{CORLT}(\mathbf{P})$<br>16 $\text{corLT}[\mathbf{P}] := \text{true}$<br>17 <b>return</b> $u_{\mathbf{P}} := \text{CORR}(\mathbf{P})$<br><br><b>Oracle</b> $\text{SEND}(\mathbf{P}, i, \text{ssid}, m, \text{role}, \text{type})$<br>18 ... // Omitted code is the same as $\mathbf{G}_4$<br>19 <b>Case</b> $\text{role} = \text{I}$ : // InitI<br>20 $(B, (S, \sigma), Y) := m_{\text{bundle}}$<br>21 $A := \mathbf{P}, A := \text{ipk}[\mathbf{P}], B := \text{ipk}[\mathbf{B}]$<br>22 <b>if</b> $\text{Verify}(B, \sigma, S) \neq \text{true}$ : <b>return</b> $\perp$<br>23 <b>if</b> $\neg \text{corLT}[\mathbf{B}] \wedge (S, \sigma) \notin \text{sspk}[\mathbf{B}, \cdot]$<br>24 $(\mathbf{P}^*, \sigma^*, S^*) := (B, \sigma, S)$<br>25 <b>abort and output</b> $(\mathbf{P}^*, \sigma^*, S^*)$<br>26 ... |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Fig. 18.** Reduction  $\mathcal{F}$  for bounding  $\mathbf{G}_3$  and  $\mathbf{G}_4$  in the proof of  $\mathbf{G}_4$ . Omitted oracles are the same as in  $\mathbf{G}_4$ .

forgery against the  $\text{EUF}^c\text{-1CMA-DDH}$  game. The number of queries made by  $\mathcal{F}$  is straightforward to analyze. Therefore, we have

$$|\Pr[\mathbf{G}_3^A = 1] - \Pr[\mathbf{G}_4^A = 1]| \leq \Pr[\text{ForgeSS}] \leq \text{Adv}_{N_{\text{LT}}, N_{\text{SS}}, Q_{\text{LT}}, 4Q_{\text{kdf}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{F})$$

**Game  $\mathbf{G}_5$ :** In  $\mathbf{G}_5$ , we simulate the signatures using the simulation algorithm  $\text{SIMsign}$  of  $\Sigma$ . We make the random oracle RO of  $\Sigma$  explicit in this game. One can construct a straightforward distinguisher between  $\mathbf{G}_4$  and  $\mathbf{G}_5$  that breaks the  $\delta_{\text{sim}}$ -simulatability of  $\Sigma$  on each party long-term public key (by Definition 9, the distinguisher also has the secret key). Hence, by applying a hybrid argument on all  $N_{\text{LT}}$  long-term public keys, we have

$$|\Pr[\mathbf{G}_4^A = 1] - \Pr[\mathbf{G}_5^A = 1]| \leq N_{\text{LT}} \cdot \delta_{\text{sim}},$$

where each hybrid distinguisher takes  $n_{\text{SS}}$  signatures.

**Game  $\mathbf{G}_6$ :** We introduce an abort event **QueryKDF** to  $\mathbf{G}_6$ . During  $\mathbf{G}_6$ , if  $\mathcal{A}$  queries  $\text{KDF}_f$  or  $\text{KDF}_r$  with correct DH tuples and the query makes  $\text{cleanDH}$  (cf. Figure 17) output a non- $\perp$  value, then we say **QueryKDF** happens and the game aborts and outputs a random bit.

We first explain the detail of  $\text{cleanDH}$ .  $\text{cleanDH}$  takes as input  $(A, X, B, S, Y)$ , and attempts to retrieve the indexes of these DH public keys. If such a search fails, then  $\text{cleanDH}$  simply ignores it and continues the search for other DH public keys. After finishing the search,  $\text{cleanDH}$  uses these indexes to determine whether their corresponding secret keys are compromised or not. Depending on the combination of the uncompromised keys,  $\text{cleanDH}$  outputs an index  $d \in \{1, 2, 3, 4\}$  that indicates which DH combination is “clean” (i.e., its corresponding secret

key is not compromised) or  $\perp$  if there are no clean DH combinations or no such indexes (which means that these DH public keys are from the adversary).

This auxiliary procedure is used to identify which type of attack the adversary is performing. For example, on input  $(A, X, B, S, Y)$ , `cleanDH` tries to find parties A and B, a semi-static key index `ssid`, and two session indexes  $i$  and  $j$  such that A and B are the long-term public keys of A and B, respectively,  $S$  is the `ssid`-th semi-static public key of B,  $X$  is the ephemeral public key of the  $i$ -th session of A, and  $Y$  is the ephemeral public key of the  $j$ -th session of B. If neither A's long-term secret key nor B's `ssid`-th semi-static secret key has been corrupted, then `cleanDH` outputs 1, which indicates that the adversary has found  $\text{CDH}(A, S)$  without launching trivial attacks (e.g., corrupting A or revealing the randomness of  $\pi_j^B$ ). The return index corresponds to the index of DH combination in the  $\text{KDF}_f$  or  $\text{KDF}_r$  query that calls `cleanDH`.

Then, we construct a **CorrGapCDH** reduction  $\mathcal{B}_{\text{dh}}$  to bound the probability of **QueryKDF**. The code of  $\mathcal{B}_{\text{dh}}$  is presented in Figure 19.  $\mathcal{B}_{\text{dh}}$  takes  $(N_{\text{LT}} + N_{\text{SS}} := N_{\text{LT}} \cdot n_{\text{SS}}) + N_{\text{ses}}$  challenge DH public keys as input, and has access to the DDH and CORR oracles. For the sake of presentation, we re-arrange the indexes of the challenge input to distinguish the long-term public keys, semi-static public keys, and ephemeral public keys. These challenge DH public keys are embedded in all honestly-generated public keys in the game. When  $\mathcal{A}$  wants to corrupt any of these keys,  $\mathcal{B}_{\text{dh}}$  simply forwards the corruption query to its CORR oracle with the corresponding index. In  $\text{KDF}_f$  and  $\text{KDF}_r$ ,  $\mathcal{B}_{\text{dh}}$  uses the DDH oracle to check if the DH tuples in the query are correct, without knowing the secret keys.

Finally, we prove that if **QueryKDF** happens, then  $\mathcal{B}_{\text{dh}}$  finds a valid solution for its **CorrGapCDH** challenge. We take the  $\text{KDF}_f$  as an example. On a query  $(Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma)$ ,  $\text{KDF}_f$  calls `cleanDH` only if all the DH tuples are correct (i.e.,  $Z_1 = \text{cdh}(A, S)$ ,  $Z_2 = \text{cdh}(X, B)$ ,  $Z_3 = \text{cdh}(X, S)$ , and  $Z_4 = \text{cdh}(X, Y)$ ). Then, if this query triggers **QueryKDF**, it means that `cleanDH` $(A, X, B, S, Y)$  outputs a non- $\perp$  value. There are four cases to consider:

- `cleanDH outputs 1`: This means that  $\mathcal{A}$  has found  $Z_1 = \text{CDH}(A, S)$ , where  $A$  is the long-term public key of an uncorrupted party and  $S$  is an uncorrupted semi-static public key. Hence,  $\mathcal{B}_{\text{dh}}$  can output  $Z_1$  as a valid solution to its **CorrGapCDH** challenge.
- `cleanDH outputs 2`: This means that  $\mathcal{A}$  has found  $Z_2 = \text{CDH}(X, B)$ , where  $X$  is the ephemeral public key of some uncompromised initiator session and  $B$  is an uncorrupted long-term public key, so  $\mathcal{B}_{\text{dh}}$  can output  $Z_2$  as a valid solution to its **CorrGapCDH** challenge.
- `cleanDH outputs 3 or 4`: Similarly,  $\mathcal{A}$  has found  $Z_3 = \text{CDH}(X, S)$  or  $Z_4 = \text{CDH}(X, Y)$ , where  $X$  is the ephemeral public key of some uncompromised initiator session and  $S$  or  $Y$  is an uncorrupted semi-static or one-time pre-key of some responder session, so  $\mathcal{B}_{\text{dh}}$  can output  $Z_3$  or  $Z_4$  as a valid solution to its **CorrGapCDH** challenge.

Therefore, if **QueryKDF** happens, then  $\mathcal{B}_{\text{dh}}$  can solve its **CorrGapCDH** challenge, and we have

$$|\Pr[\mathbf{G}_5^{\mathcal{A}} = 1] - \Pr[\mathbf{G}_6^{\mathcal{A}} = 1]| \leq \Pr[\mathbf{QueryKDF}]$$

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Game <math>\mathcal{B}_{\text{dh}}^{\text{DDH, CORR}}</math></b> <math>\left( \begin{array}{l} (U_i)_{1 \leq i \leq N_{\text{LT}}}, \\ (S_{i,j})_{(i,j) \in [N_{\text{LT}}] \times [n_{\text{SS}}]}, \\ (R_i)_{1 \leq i \leq N_{\text{see}}} \end{array} \right)</math></p> <pre> 01 <math>\beta \xleftarrow{\\$} \{0, 1\}</math>, <math>\text{cnt} := 0</math>, <math>Z^* \xleftarrow{\\$} \mathbb{G}</math> 02 <b>for</b> <math>P \in [N_{\text{LT}}]</math> 03   <math>(\text{isk}[P], \text{ipk}[P]) := (\perp, U_P)</math> 04   <b>for</b> <math>\text{ssid} \in [n_{\text{SS}}]</math> 05     <math>S := S_{P, \text{ssid}}</math> // <math>((P-1) \cdot n_{\text{SS}} + \text{ssid})</math>-th <math>S</math> 06     <math>(\sigma, q, h) \xleftarrow{\\$} \text{SIMsign}(U_P, S)</math> 07     <math>\mathcal{L}_{\text{RO}} := \mathcal{L}_{\text{RO}} \cup \{(q, h)\}</math> 08     <math>\text{sspk}[P, \text{ssid}] := \perp</math> 09     <math>\text{sspk}[P, \text{ssid}] := (S, \sigma)</math> 10 <math>\beta^* \leftarrow A^O(\text{ipk}, \text{sspk})</math> 11 // Output a solution if <math>\text{QueryKDF}</math> never occurs 12 <b>return</b> <math>Z^*</math>  <b>Oracle DER</b>(<math>P, i, \text{ssid}, m</math>) 13 <math>\pi_i^P := \Pi[P, i]</math> 14 <b>if</b> <math>\pi_i^P = \perp \vee \pi_i^P.\text{SK} \neq \perp</math>: <b>return</b> <math>\perp</math> 15 <math>B := P, (b, B) := (\text{isk}[P], \text{ipk}[P])</math> 16 <math>(\perp, Y) := \pi.\text{state}, (S, \sigma) := \text{sspk}[P, \text{ssid}]</math> 17 <math>(A, X) := m, A := \text{ipk}[A]</math> 18 <math>K := \text{PatchSK}(A, X, B, S, Y, \sigma)</math> 19 <math>\pi.\text{pid}, \text{SK} := (A, K)</math> 20 <math>\pi.\text{sid} := (A, X, B, S, Y, \sigma)</math> 21 <math>\Pi[P, i] := \pi_i^P</math>; <b>return</b> <math>\epsilon</math>  <b>Oracle KDF<sub>r</sub></b>(<math>Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma</math>) 22 <b>if</b> <math>\text{DDH}(A, S, Z_1)</math> // <math>Z_1 = \text{cdh}(A, S)</math> 23   <math>\wedge \text{DDH}(X, B, Z_2)</math> // <math>Z_2 = \text{cdh}(X, B)</math> 24   <math>\wedge \text{DDH}(X, S, Z_3)</math> // <math>Z_3 = \text{cdh}(X, S)</math> 25   <math>\wedge \text{DDH}(X, Y, Z_4)</math> // <math>Z_4 = \text{cdh}(X, Y)</math> 26   <b>if</b> <math>d := \text{cleanDH}(A, X, B, S, Y) \neq \perp</math> 27     <b>abort and output</b> <math>Z_d</math> 28 ... // The rest is the same as <math>\mathbb{G}_6</math>  <b>Oracle KDF<sub>r</sub></b>(<math>Z_1, Z_2, Z_3, A, X, B, S, \sigma</math>) 29 <b>if</b> <math>\text{DDH}(A, S, Z_1)</math> // <math>Z_1 = \text{cdh}(A, S)</math> 30   <math>\wedge \text{DDH}(X, B, Z_2)</math> // <math>Z_2 = \text{cdh}(X, B)</math> 31   <math>\wedge \text{DDH}(X, S, Z_3)</math> // <math>Z_3 = \text{cdh}(X, S)</math> 32   <b>if</b> <math>d := \text{cleanDH}(A, X, B, S, \perp) \neq \perp</math> 33     <b>abort and output</b> <math>Z_d</math> 34 ... // The rest is the same as <math>\mathbb{G}_6</math>  <b>Oracle CORLT</b>(<math>P</math>) 35 <math>\text{corLT}[P] := \text{true}</math> 36 <b>return</b> <math>\text{CORR}(P)</math></pre> | <p><b>Oracle CORSS</b>(<math>P, \text{ssid}</math>)</p> <pre> 37 <math>\text{corSS}[\text{sspk}[P, \text{ssid}]] := \text{true}</math> 38 // <math>S_i</math> is the <math>\text{ssid}</math>-th semi-static pk of <math>P</math> 39 <math>i := (N_{\text{LT}} + (P-1) \cdot n_{\text{SS}} + \text{ssid})</math> 40 <b>return</b> <math>\text{CORR}(i)</math>  <b>Oracle REVRAND</b>(<math>P, i</math>) 41 <b>if</b> <math>\pi_i^P.\text{rand} \neq \perp</math> 42   <math>\text{revRand}[\pi_i^P] := \text{true}</math> 43   <math>k := \pi_i^P.\text{rand}</math> 44   <b>return</b> <math>\text{CORR}(N_{\text{LT}} + N_{\text{SS}} + k)</math> 45 <b>return</b> <math>\perp</math>  <b>Oracle SEND</b>(<math>P, i, \text{ssid}, m, \text{role}, \text{type}</math>) 46 ... // Setup variables of <math>\pi</math> 47 <math>\text{cnt} := \text{cnt} + 1</math> 48 <math>\pi.\text{rand} := \text{cnt}</math> // For indexing random coins 49 <b>Case</b> <math>\text{role} = \text{R}</math> // InitR 50   <math>B := P, B := \text{ipk}[P]</math> 51   <math>(S, \sigma) := \text{sspk}[P, \text{ssid}]</math> 52   <b>if</b> <math>\text{type} = \text{full}</math>: <math>Y := R_{\text{cnt}}</math> 53   <b>else</b> <math>Y := \perp</math> 54   <math>m' := m_{\text{bundle}} := (B, (S, \sigma), Y)</math> 55   ... // Update variables of <math>\pi</math> 56 <b>Case</b> <math>\text{role} = \text{I}</math>: // InitI 57   <math>(B, (S, \sigma), Y) := m_{\text{bundle}}</math> 58   <math>A := P, A := \text{ipk}[P], B := \text{ipk}[B]</math> 59   <b>if</b> <math>\text{Verify}(B, \sigma, S) \neq \text{true}</math>: <b>return</b> <math>\perp</math> 60   <b>else</b> <math>\pi.\text{sspk} := (S, \sigma)</math> 61   <b>if</b> <math>\neg \text{corLT}[B] \wedge (S, \sigma) \notin \text{sspk}[B, \cdot]</math> 62     <b>abort and output</b> <math>Z^*</math> 63   <b>if</b> <math>Y \neq \perp</math>: <math>\pi.\text{type} := \text{full}</math> 64   <b>else</b> <math>\pi.\text{type} := \text{reduced}</math> 65   <math>X := R_{\text{cnt}}, m' := (A, X)</math> 66   <math>K := \text{PatchSK}(A, X, B, S, Y, \sigma)</math> 67   ... // Update variables of <math>\pi</math> 68 <b>return</b> <math>m'</math>  <b>Procedure PatchSK</b>(<math>A, X, B, S, Y, \sigma</math>) 69 <b>if</b> <math>\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] = \perp</math> 70   <math>K \xleftarrow{\\$} \text{SK}</math> 71   <math>\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] := K</math> 72 <b>return</b> <math>\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma]</math>  <b>Procedure cleanDH</b>(<math>A, X, B, S, Y</math>) 73 ... 74 <b>Find</b> <math>(i, k)</math> s.t. <math>k = \Pi[A, i].\text{rand} \wedge X = R_k</math> 75 <b>Find</b> <math>(j, \ell)</math> s.t. <math>\ell = \Pi[B, j].\text{rand} \wedge Y = R_\ell</math> 76 ... // The rest is the same as <math>\text{cleanDH}</math> in Figure 17</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Fig. 19.** Reduction  $\mathcal{B}_{\text{dh}}$  for bounding the transition from  $\mathbb{G}_5$  to  $\mathbb{G}_6$ . Omitted oracles are the same as in  $\mathbb{G}_5$ .

$$\leq \text{Adv}_{N_{\text{LT}}+N_{\text{SS}}+N_{\text{ses}}, Q_{\text{LT}}+Q_{\text{SS}}+Q_{\text{Eph}}, 4Q_{\text{kdf}}}^{\text{CorrGapCDH}}(\mathcal{B}_{\text{dh}})$$

ANALYSIS OF  $\mathbf{G}_6$ . Now we claim that the winning probability of  $\mathcal{A}$  in  $\mathbf{G}_6$  is  $1/2$ . To see this, we first observe that, in  $\mathbf{G}_1$ , all public keys are unique, so each full session has unique **sid** and only responder sessions in the reduced mode could have repeated **sid**'s. Hence, **Sound()** will never output **false**, and  $\mathcal{A}$  can only win by launching valid attacks on fresh sessions.

We first prove that, if a completed session  $\pi_i^P$  is fresh, then  $\pi.\text{SK}$  is distributed uniformly at random in  $\mathcal{A}$ 's view. Let  $\pi$  be an arbitrary test session with **sid** =  $(A, X, B, S, Y, \sigma)$  and let the session be in the full mode (similar analysis applies to the reduced mode). In  $\mathbf{G}_6$ ,  $\pi.\text{SK}$  is sampled by **PatchSK**( $A, X, B, S, Y, \sigma$ ). In the random oracle model, the only way for  $\mathcal{A}$  to distinguish whether  $K$ , returned by **TEST**( $P, i$ ), is uniformly random or generated by **PatchSK** is to query  $\text{KDF}_f$  on the exact query  $(Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma)$ , which causes  $\text{KDF}_f$  to invoke **PatchSK**. If  $\mathcal{A}$  never issues such a query, then  $K \leftarrow \text{PatchSK}(A, X, B, S, Y, \sigma)$  has the same distribution as  $K \leftarrow^{\$} \text{SK}$  in  $\mathcal{A}$ 's view. Note that revealing  $\pi$  also can distinguish them, but then it violates the freshness of  $\pi$ .

Therefore, if  $\mathcal{A}$  can win with probability non-negligibly larger than  $1/2$ , then  $\mathcal{A}$  must query  $\text{KDF}_f$  on the exact query  $(Z_1, Z_2, Z_3, Z_4, A, X, B, S, Y, \sigma)$  of an arbitrary session fresh  $\pi$ , where  $Z_1 = \text{cdh}(A, S)$ ,  $Z_2 = \text{cdh}(X, B)$ ,  $Z_3 = \text{cdh}(X, S)$ ,  $Z_4 = \text{cdh}(X, Y)$ ,  $A$  and  $B$  are the long-term public keys of  $\pi$ 's initiator and responder, respectively,  $X$  is the initiator ephemeral public key, and  $\pi$ 's pre-key bundle is  $(B, (S, \sigma), Y)$ . By the freshness definition of  $\text{X3DH}$ , we have  $\text{clean}_{\text{full}}^{\text{X3DH}}(\pi) = \text{true}$ . By the definition of  $\text{clean}_{\text{full}}^{\text{X3DH}}$ , at least one of the following predicates outputs **true**: **LT-SS**, **Eph-LT**, **Eph-SS**, and **Eph-Eph**. We analyze all these predicates as follows:

- *Case* **LT-SS**( $\pi$ ) = **true**. Let  $A$  and  $B$  be the initiator and responder in  $\pi$ , respectively, and **ssid** be the semi-static key index of  $B$  used in  $\pi$ . Under this case,  $A$  is uncorrupted (i.e., **corLT**[ $A$ ] = **false**) and  $B$ 's **ssid**-th semi-static key is secure. More specifically:
  - If  $\pi$  is a responder session (i.e., owned by  $B$ ), then **sspk**[ $B, \text{ssid}$ ] is also uncorrupted (i.e., **corSS**[**sspk**[ $B, \text{ssid}$ ]] = **false**). Hence, if  $\mathcal{A}$  issues such a RO query to  $\text{KDF}_f$ , then **cleanDH** will output 1 and trigger **QueryKDF**.
  - If  $\pi$  is an initiator session (i.e., owned by  $A$ ), then **sspk**[ $B, \text{ssid}$ ] is uncorrupted and either (1)  $B$  was uncorrupted when  $\pi$  accepted a key, or (2)  $(\text{sspk}, \sigma)$  is one of  $B$ 's semi-static keys. In  $\mathbf{G}_4$ , if  $B$  was uncorrupted when  $\pi$  accepted a key, then excluding **ForgeSS** ensures that  $(\text{sspk}, \sigma)$  must come from  $B$ . Hence, subcases (1) and (2) collapse into (2), which means that **cleanDH** will find a **ssid** such that **sspk**[ $B, \text{ssid}$ ] =  $(S, \sigma)$ . Therefore, **corSS**[ $(S, \sigma)$ ] = **false**, and **cleanDH** will output 1 and trigger **QueryKDF**.
- *Case* **Eph-SS**( $\pi$ ) = **true**. This is similar to the previous case: the randomness of the related initiator session of  $\pi$  (either  $\pi$  itself or its matching session, let this session be  $\pi_i^P$ ) is not revealed and the used semi-static key is uncorrupted (by the argument in the previous case). Hence, **revRand**[ $P, i$ ] = **false** and

- $\text{corSS}[(S, \sigma)] = \text{false}$ , and  $\text{cleanDH}$  will output 3 and trigger **QueryKDF**.
- *Case Eph-LT*( $\pi$ ) = **true**. In this case, the randomness of the related initiator session (let it be  $\pi_i^A$ ) of  $\pi$  is not revealed (i.e.,  $\text{revRand}[A, i] = \text{false}$ ) and  $\text{corLT}[B] = \text{false}$ , which makes  $\text{cleanDH}$  output 2 and trigger **QueryKDF**.
  - *Case Eph-Eph*( $\pi$ ) = **true**. Let  $\pi_i^A$  and  $\pi_j^B$  be the related initiator session and related responder session of  $\pi$  (either itself or its matching session), respectively. Then, in this case, we have  $\text{revRand}[A, i] = \text{false} = \text{revRand}[B, j]$ , which makes  $\text{cleanDH}$  output 4 and trigger **QueryKDF**.

Therefore, excluding **QueryKDF** ensures that  $\mathcal{A}$  can never query  $\text{KDF}_f$  or  $\text{KDF}_r$  on the exact query of any full or reduced session that is fresh, which means that the session key of any fresh session is indistinguishable from random to  $\mathcal{A}$ . Hence,  $\mathcal{A}$  can only win by guessing the bit in the test query, and we have

$$\Pr [\mathbf{G}_6^{\mathcal{A}} = 1] \leq \frac{1}{2}.$$

By combining all probability bounds in the previous games, we have

$$\begin{aligned} \left| \Pr [\text{BAKE}_{\text{X3DH}^+}^{\mathcal{A}}(\mathcal{A}) = 1] - \frac{1}{2} \right| &\leq \text{Adv}_{N_{\text{LT}}, N_{\text{SS}}, Q_{\text{LT}}, 4Q_{\text{kdf}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{F}) \\ &\quad + \text{Adv}_{N_{\text{LT}} + N_{\text{SS}} + N_{\text{ses}}, Q_{\text{LT}} + Q_{\text{SS}} + Q_{\text{Eph}}, 4Q_{\text{kdf}}}^{\text{CorrGapCDH}}(\mathcal{B}_{\text{dh}}) \\ &\quad + N_{\text{LT}} \cdot \delta_{\text{sim}} + \frac{(N_{\text{LT}} + N_{\text{SS}} + N_{\text{ses}})^2}{2p} \end{aligned}$$

## C Security Proof of XHMQV (Theorem 2)

The security proof of XHMQV has a similar structure as the proof of X3DH in Appendix B, but with some important differences: (1) The proof of XHMQV requires the simulation of random oracles  $H_0, H_1, H_2$ , and  $H_3$  because they are related to the  $\text{CorrCRGapCDH}$  assumption (cf. Figure 5). (2) The DH solution extractor of the proof of XHMQV has a more complicated structure because the XHMQV's DH shared secret has more “twists” than X3DH. We present the game sequence in Figure 20. Since the game sequence  $\mathbf{G}_0$ - $\mathbf{G}_5$  in the proof of XHMQV performs similar changes as the game sequence in the proof of X3DH, we skip them and directly present the game  $\mathbf{G}_6$  where the simulation of random oracles and the DH solution extractor are introduced. The bounds of these games are the same as in the proof of X3DH. Therefore, we have

$$\begin{aligned} &\left| \Pr [\text{BAKE}_{\text{XHMQV}}^{\mathcal{A}}(\mathcal{A}) = 1] - \Pr [\mathbf{G}_5^{\mathcal{A}} = 1] \right| \\ &\leq \frac{(N_{\text{LT}} + N_{\text{SS}} + N_{\text{ses}})^2}{2p} + \text{Adv}_{N_{\text{LT}}, N_{\text{SS}}, Q_{\text{LT}}, Q_{\text{kdf}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{F}) + N_{\text{LT}} \cdot \delta_{\text{sim}}, \end{aligned}$$

**Game  $\mathbf{G}_6$ :** An abort event **QueryKDF** is introduced to  $\mathbf{G}_6$ . During  $\mathbf{G}_6$ , if  $\mathcal{A}$  queries  $\text{KDF}_f$  or  $\text{KDF}_r$  with correct XHMQV DH secret and the query makes

| Game $G_0-G_6$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Oracle $\text{SEND}(P, i, \text{ssid}, m, \text{role}, \text{type})$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01 $\beta \xleftarrow{\$} \{0, 1\}$<br>02 <b>for</b> $P \in [N_{LT}]$<br>03 $(\text{isk}[P], \text{ipk}[P]) := (u, U) \leftarrow \text{KGenLT}$<br>04 <b>for</b> $\text{ssid} \in [n_{SS}]$<br>05 $(s, (S, \sigma)) \leftarrow \text{KGenSS}(u, U)$<br>06 $(\sigma, q, h) \xleftarrow{\$} \text{SIMsign}(U, S)$ // $G_5-G_6$<br>07 $\mathcal{L}_{RO} := \mathcal{L}_{RO} \cup \{(q, h)\}$ // $G_5-G_6$<br>08 $(\text{sssk}[P, \text{ssid}], \text{sspk}[P, \text{ssid}]) := (s, (S, \sigma))$<br>09 $\beta^* \leftarrow \mathcal{A}^O(\text{ipk}, \text{sspk})$<br>10 <b>if</b> $\text{Sound}() = \text{false}$ : <b>return</b> 1<br>11 <b>if</b> $\exists \pi \in \mathcal{S}_{\text{test}} \text{ s.t. } \text{Fresh}(\pi) = \text{false}$ : <b>return</b> 0<br>12 <b>return</b> $\llbracket \beta = \beta^* \rrbracket$                                                                                                                                                                                                                                 | 43 <b>if</b> $\pi := \Pi[P, i] \neq \perp$ : <b>return</b> $\perp$<br>44 <b>if</b> $\text{role} \notin \{I, R\}$<br>45 $\vee \text{type} \notin \{\text{full}, \text{reduced}\}$<br>46 <b>return</b> $\perp$<br>47 $r \xleftarrow{\$} \mathbb{Z}_p^*$<br>48 $\pi.(\text{oid}, \text{role}, \text{rand}) := (P, \text{role}, r)$<br>49 <b>Case</b> $\text{role} = R$ // $\text{InitR}$<br>50 $B := P, (b, B) := (\text{isk}[P], \text{ipk}[P])$<br>51 $s := \text{sssk}[P, \text{ssid}]$<br>52 $(S, \sigma) := \text{sspk}[P, \text{ssid}]$<br>53 <b>if</b> $\text{type} = \text{full}$ : $y := r, Y := g^y$<br>54 <b>else</b> $Y := \perp$<br>55 $m' := m_{\text{bundle}} := (B, (S, \sigma), Y)$<br>56 $\pi.(\text{pid}, \text{type}) := (*, \text{type})$<br>57 $\pi.(\text{ssk}, \text{state}) := ((s, (S, \sigma)), (y, Y))$<br>58 $\pi.\text{cid} := (B, (S, \sigma), Y)$<br>59 <b>Case</b> $\text{role} = I$ : // $\text{InitI}$<br>60 $A := P, (B, (S, \sigma), Y) := m_{\text{bundle}}$<br>61 $(a, A) := (\text{isk}[A], \text{ipk}[A]), B := \text{ipk}[B]$<br>62 <b>if</b> $\text{Verify}(B, \sigma, S) \neq \text{true}$<br>63 <b>return</b> $\perp$<br>64 <b>else</b> $\pi.\text{sspk} := (S, \sigma)$<br>65 <b>if</b> $\neg \text{corLT}[B]$ // $G_4-G_6$<br>66 $\wedge (S, \sigma) \notin \text{sspk}[B, \cdot]$ // $G_4-G_6$<br>67 <b>abort and output</b> $\beta$ // $G_4-G_6$<br>68 <b>if</b> $Y \neq \perp$ : $\pi.\text{type} := \text{full}$<br>69 <b>else</b> $\pi.\text{type} := \text{reduced}$<br>70 $x := \pi.\text{rand}, X := g^x$<br>71 $m' := (A, X)$<br>72 $d := H_0(A, X)$<br>73 <b>if</b> $\pi.\text{type} = \text{full}$<br>74 $e_1 := H_1(B, Y, S), e_2 := H_2(B, Y, S)$<br>75 $\text{DH} := (YB^{e_1}S^{e_2})^{x+da}$<br>76 $\text{ctxtr} := (A, X, B, S, Y, \sigma)$<br>77 $K := \text{KDF}_f(\text{DH}, \text{ctxtr})$<br>78 <b>else</b><br>79 $e_1 := H_3(B, S)$<br>80 $\text{ctxtr} := (A, X, B, S, \sigma)$<br>81 $\text{DH} := (B^{e_1}S)^{x+da}$<br>82 $K := \text{KDF}_f(\text{DH}, \text{ctxtr})$<br>83 $K := \text{PatchSK}(A, X, B, S, Y, \sigma)$ // $G_3-G_6$<br>84 $\pi.(\text{pid}, \text{SK}) := (B, K)$<br>85 $\pi.\text{cid} := (B, (S, \sigma), Y)$<br>86 $\pi.\text{sid} := (A, X, B, S, Y, \sigma)$<br>87 $\text{pcorLT}[\pi_i^P] := \text{corLT}[\pi_i^P.\text{pid}]$<br>88 $\Pi[P, i] := \pi_i^P$<br>89 <b>return</b> $m'$ |
| <b>Oracle</b> $\text{DER}(P, i, \text{ssid}, m)$<br>13 $\pi_i^P := \Pi[P, i]$<br>14 <b>if</b> $\pi_i^P = \perp \vee \pi_i^P.\text{SK} \neq \perp$ : <b>return</b> $\perp$<br>15 $B := P, (b, B) := (\text{isk}[P], \text{ipk}[P])$<br>16 $(s, (S, \sigma)) := (\text{sssk}[P, \text{ssid}], \text{sspk}[P, \text{ssid}])$<br>17 $(y, Y) := \pi.\text{state}$<br>18 $(A, X) := m, A := \text{ipk}[A]$<br>19 <b>if</b> $\pi.\text{type} = \text{full}$<br>20 $e_1 := H_1(B, Y, S), e_2 := H_2(B, Y, S)$<br>21 $\text{DH} := (XA^d)^{y+e_1b+e_2s}$<br>22 $\text{ctxtr} := (A, X, B, S, Y, \sigma)$<br>23 $K := \text{KDF}_f(\text{DH}, \text{ctxtr})$<br>24 <b>else</b><br>25 $e_1 := H_3(B, S)$<br>26 $\text{ctxtr} := (A, X, B, S, \sigma)$<br>27 $\text{DH} := (XA^d)^{e_1b+s}$<br>28 $K := \text{KDF}_f(\text{DH}, \text{ctxtr})$<br>29 $K := \text{PatchSK}(A, X, B, S, Y, \sigma)$ // $G_3-G_6$<br>30 $\pi.(\text{pid}, \text{SK}) := (A, K)$<br>31 $\pi.\text{sid} := (A, X, B, S, Y, \sigma)$<br>32 $\Pi[P, i] := \pi_i^P$ ; <b>return</b> $\epsilon$ |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Oracle</b> $\text{KDF}_f(Z, A, X, B, S, Y, \sigma)$<br>33 $d := H_0(A, X)$ // $G_2-G_6$<br>34 $e_1 := H_1(B, Y, S), e_2 := H_2(B, Y, S)$ // $G_2-G_6$<br>35 <b>if</b> $Z = \text{cdh}(XA^d, YB^{e_1}S^{e_2})$ // $G_2-G_6$<br>36 <b>if</b> $\text{cleanHMQV}(A, X, B, S, Y, Z) \neq \perp$ // $G_6$<br>37 <b>abort and output</b> $\beta$ // $G_6$<br>38 <b>return</b> $\text{PatchSK}(A, X, B, S, Y, \sigma)$ // $G_2-G_6$<br>39 <b>if</b> $\mathcal{L}_{\text{KDF}_f}[Z, A, X, B, S, Y, \sigma] = \perp$<br>40 $K \xleftarrow{\$} \mathcal{SK}$<br>41 $\mathcal{L}_{\text{KDF}_f}[Z, A, X, B, S, Y, \sigma] := K$<br>42 <b>return</b> $\mathcal{L}_{\text{KDF}_f}[Z, A, X, B, S, Y, \sigma]$                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

**Fig. 20.** Games  $G_0-G_6$  in proving Theorem 2. The oracles  $\text{CORLT}$ ,  $\text{CORSS}$ ,  $\text{REVRAND}$ ,  $\text{REVSK}$  and  $\text{TEST}$  and auxiliary procedures  $\text{Fresh}$  and  $\text{Sound}$  are the same as in Figure 14. The auxiliary procedure  $\text{cleanHMQV}$  (for  $G_6$ ) and random oracles  $\text{RO}$ ,  $H_0$ ,  $H_1$ ,  $H_2$ , and  $H_3$  are defined in Figure 21.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Procedure</b> $\text{cleanHMQV}(A, X, B, S, Y, Z)$<br>01 $d := 0$<br>02 Find $\pi$ such that $\pi.\text{sid} = (A, X, B, S, Y, \cdot)$<br>03 <b>if</b> $\pi = \perp$<br>04 <b>return</b> $\perp$ // No corresponding sessions<br>05 Find $A$ s.t. $\text{ipk}[A] = A$<br>06 Find $B$ s.t. $\text{ipk}[B] = B$<br>07 Find $\text{ssid}$ s.t. $\text{sspk}[B, \text{ssid}] = (S, \cdot)$<br>08 Find $i$ s.t. $x = \Pi[A, i].\text{rand}$ and $X = g^x$<br>09 Find $j$ s.t. $y = \Pi[B, j].\text{rand}$ and $Y = g^y$<br>10 <b>if</b> $i \neq \perp$<br>11 $\text{role} := \text{I}$ // $X$ is under A's control<br>12 <b>elseif</b> $j \neq \perp \wedge \text{ssid} \neq \perp$<br>13 $\text{role} := \text{R}$ // $S$ and $Y$ are under B's control<br>14 <b>else return</b> $\perp$ // No corresponding sessions<br>15 <b>if</b> $\text{ssid} \neq \perp \wedge \neg \text{corLT}[A] \wedge \neg \text{corSS}[B, \text{ssid}]$<br>16 $d := 1$ // Detect a valid LT-SS attack<br>17 <b>if</b> $i \neq \perp \wedge \neg \text{revRand}[A, i] \wedge \neg \text{corLT}[B]$<br>18 $d := 2$ // Detect a valid Eph-LT attack<br>19 <b>if</b> $i \neq \perp \wedge \text{ssid} \neq \perp$<br>20 $\wedge \neg \text{revRand}[A, i] \wedge \neg \text{corSS}[B, \text{ssid}]$<br>21 $d := 3$ // Detect a valid Eph-SS attack<br>22 <b>if</b> $i \neq \perp \wedge j \neq \perp$<br>23 $\wedge \neg \text{revRand}[A, i] \wedge \neg \text{revRand}[B, j]$<br>24 $d := 4$ // Detect a valid Eph-Eph attack<br>25 <b>if</b> $\neg \text{corLT}[A] \wedge \neg \text{corLT}[B]$<br>26 $d := 5$ // Detect a valid LT-LT attack<br>27 <b>if</b> $j \neq \perp \wedge \neg \text{corLT}[A] \wedge \neg \text{revRand}[B, j]$<br>28 $d := 6$ // Detect a valid LT-Eph attack<br>29 <b>if</b> $d = 0$ : <b>return</b> $\perp$<br>30 $\text{type} := \pi.\text{type}$ // Full or reduced mode<br>31 Compute $Z^*$ as in Table 2<br>based on the variables $\text{role}, \text{type}$ and $d$<br>32 <b>return</b> $Z^*$<br><br><b>Oracle</b> $\text{RO}(q)$ // Used by $\Sigma$<br>33 <b>if</b> $\mathcal{L}_{\text{RO}}[q] = \perp$<br>34 $h \xleftarrow{\$} \text{RO.image}$<br>35 $\mathcal{L}_{\text{RO}}[q] := h$<br>36 <b>return</b> $\mathcal{L}_{\text{RO}}[q]$ | <b>Oracle</b> $H_0(A, X)$<br>37 <b>if</b> $\mathcal{L}_{H_0}[A, X] = \perp$<br>38 $h \xleftarrow{\$} \mathbb{Z}_p$<br>39 $\mathcal{L}_{H_0}[A, X] := h$<br>40 <b>return</b> $\mathcal{L}_{H_0}[A, X]$<br><br><b>Oracle</b> $H_1(B, Y, S)$<br>41 <b>if</b> $\mathcal{L}_{H_1}[B, Y, S] = \perp$<br>42 $h \xleftarrow{\$} \mathbb{Z}_p$<br>43 $\mathcal{L}_{H_1}[B, Y, S] := h$<br>44 <b>return</b> $\mathcal{L}_{H_1}[B, Y, S]$<br><br><b>Oracle</b> $H_2(B, Y, S)$<br>45 <b>if</b> $\mathcal{L}_{H_2}[B, Y, S] = \perp$<br>46 $h \xleftarrow{\$} \mathbb{Z}_p$<br>47 $\mathcal{L}_{H_2}[B, Y, S] := h$<br>48 <b>return</b> $\mathcal{L}_{H_2}[B, Y, S]$<br><br><b>Oracle</b> $H_3(B, S)$<br>49 <b>if</b> $\mathcal{L}_{H_3}[B, S] = \perp$<br>50 $h \xleftarrow{\$} \mathbb{Z}_p$<br>51 $\mathcal{L}_{H_3}[B, S] := h$<br>52 <b>return</b> $\mathcal{L}_{H_3}[B, S]$<br><br><b>Oracle</b> $\text{KDF}_r(Z, A, X, B, S, \sigma)$<br>53 $d := H_0(A, X), e_1 := H_3(B, S)$ // $\mathbf{G}_2\text{-}\mathbf{G}_6$<br>54 <b>if</b> $Z = \text{cdh}(XA^d, B^{e_1}S)$ // $\mathbf{G}_2\text{-}\mathbf{G}_6$<br>55 <b>if</b> $\text{cleanHMQV}(A, X, B, S, \perp, Z) \neq \perp$ // $\mathbf{G}_6$<br>56 <b>abort</b> and <b>output</b> $\beta$ // $\mathbf{G}_6$<br>57 <b>return</b> $\text{PatchSK}(A, X, B, S, \perp, \sigma)$ // $\mathbf{G}_2\text{-}\mathbf{G}_6$<br>58 <b>if</b> $\mathcal{L}_{\text{KDFr}}[Z, A, X, B, S, \sigma] = \perp$<br>59 $K \xleftarrow{\$} SK$<br>60 $\mathcal{L}_{\text{KDFr}}[Z, A, X, B, S, \sigma] := K$<br>61 <b>return</b> $\mathcal{L}_{\text{KDFr}}[Z, A, X, B, S, \sigma]$<br><br><b>Procedure</b> $\text{PatchSK}(A, X, B, S, Y, \sigma)$<br>62 <b>if</b> $\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] = \perp$<br>63 $K \xleftarrow{\$} SK$<br>64 $\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] := K$<br>65 <b>return</b> $\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma]$ |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Fig. 21.** Random oracles  $\text{RO}$  (used by the signature  $\Sigma$ ),  $H_0, H_1, H_2$ , and  $H_3$  and auxiliary procedures  $\text{PatchSK}$  and  $\text{cleanHMQV}$  in Figure 20.

`cleanHMQV` (cf. Figure 21) output a non- $\perp$  value, then we say **QueryKDF** happens and the game aborts and outputs a random bit.

The `cleanHMQV` procedure is similar to `cleanDH` in the proof of X3DH but with more cases (LT-LT and LT-Eph) and two important differences:

- On input  $(A, X, B, S, Y, Z)$ , `cleanHMQV` first checks if the input corresponds to any session’s `sid`. If not, `cleanHMQV` outputs  $\perp$ . Otherwise, `cleanHMQV` proceeds as `cleanDH`, finding owner, peer, `ssid`, and session index of  $\pi$ . It also retrieves the type and role of  $\pi$ .

These extra steps are necessary for the proof of XHMQV. Different from X3DH, a DH extractor of XHMQV cannot directly compute the DH solution. It requires relevant information of the session (e.g., the session’s other keys) to help compute the DH solution.

- `cleanHMQV` not only determines which attack case the session corresponds to, but also computes a DH shared secret according to the attack case and Table 2.

We then explain Table 2 and the reason of using the `CorrCRGapCDH` assumption. The motivation of Table 2 is to show how a DH extractor computes a solution (or a DH shared secret that can be inverted into a solution) for `CorrCRGapCDH`. Suppose that a responder session  $\pi$  found in `cleanHMQV` is in the full mode and falls into the LT-SS case, where  $d = 1$ . Then, we have that  $Z = \text{CDH}(XA^d, YB^{e_1}S^{e_2}) = g^{(x+da) \cdot (y+e_1b+e_2s)}$ . Since  $\text{LT-SS}(\pi) = \mathbf{true}$ , a reduction should keep  $a$  and  $s$  unknown. In this case,  $b$  and  $y$  are controlled by the reduction, but  $x$  is unknown (e.g.,  $\mathcal{A}$  could try impersonating party A via a malicious  $X$ ). This causes that the known information of the reduction is not sufficient to compute  $\text{CDH}(A, S)$ , but can only compute  $\text{CDH}(XA^d, S)$  (as shown in Table 2). Other attack cases have similar issues that the known information of the reduction may not be sufficient to compute the exact CDH solutions, but some “approximate” solutions.

This is the reason why we need to use the `CorrCRGapCDH` assumption, which, informally, allows the reduction to solve CDH if it can compute some approximate solutions. In the above example,  $\text{CDH}(XA^d, S)$  is understood as an “approximate” solution of  $\text{CDH}(A, S)$ . To fit in the solution structure of `CorrCRGapCDH`, the reduction also needs to simulate the random oracles  $H_0, H_2, H_1$ , and  $H_3$  properly to make the computed “approximate” solutions satisfy the requirement of `CorrCRGapCDH`.

Now we are ready to present reductions that extract `CorrCRGapCDH` solutions from  $\mathcal{A}$ ’s queries to  $\text{KDF}_f$  or  $\text{KDF}_r$  if its queries make `cleanHMQV` output a non- $\perp$  value. As we discussed above, these reductions must simulate  $H_0, H_2, H_1$ , and  $H_3$  properly to fit in the solution structure of `CorrCRGapCDH`. However, we cannot build a single reduction for all attack cases because different cases may have different ways of simulating these random oracles. To address this, we first split

**Table 2.** Given  $(A, X, B, S, Y, Z)$ , compute  $Z^*$  based on **type**, **role**, and  $t$  in cleanHMQV of Figure 21.  $u := \text{CORR}(U)$  means that we find the index of  $U$  in the challenge input and query CORR (as in Figure 22).

| $t$ | role = I                                                                                                                                                                                                |                                                                                                                                                               | role = R                                                                                                                                                                 |                                                                                                                                                           |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | type = full                                                                                                                                                                                             | type = reduced                                                                                                                                                | type = full                                                                                                                                                              | type = reduced                                                                                                                                            |
| 1   | $x := \text{CORR}(X)$<br>$b := \text{CORR}(B)$<br>$\left(\frac{Z}{(YB^{e_1}S^{e_2})^x \cdot A^{de_1b}}\right)^{d^{-1}}$<br>$= (XA^d)^s := Z^*$                                                          | $x := \text{CORR}(X)$<br>$b := \text{CORR}(B)$<br>$\left(\frac{Z}{(SB^{e_1})^x \cdot A^{de_1b}}\right)^{d^{-1}}$<br>$= A^s := Z^*$                            | $X$ may be from $\mathcal{A}$<br>$b := \text{CORR}(B)$<br>$\left(\frac{Z}{(XA^d)^{y+e_1b}}\right)^{e_2^{-1}}$<br>$= (XA^d)^s := Z^*$                                     | $X$ may be from $\mathcal{A}$<br>$b := \text{CORR}(B)$<br>$\left(\frac{Z}{(XA^d)^{e_1b}}\right)$<br>$= A^s := Z^*$                                        |
| 2   | $Y$ may be from $\mathcal{A}$<br>$S$ may be from $\mathcal{A}$<br>$a := \text{CORR}(A)$<br>$\frac{Z}{(YB^{e_1}S^{e_2})^{ad}}$<br>$= (YS^{e_2} \cdot B^{e_1})^x := Z^*$                                  | No $Y$ involved<br>$S$ may be from $\mathcal{A}$<br>$a := \text{CORR}(A)$<br>$\frac{Z}{(SB^{e_1})^{ad}}$<br>$= (S \cdot B^{e_1})^x := Z^*$                    | $y := \text{CORR}(Y)$<br>$s := \text{CORR}(S)$<br>$a := \text{CORR}(A)$<br>$\left(\frac{Z}{(XA^d)^{y+se_2} \cdot B^{e_1ad}}\right)^{e_1^{-1}}$<br>$= B^x := Z^*$         | $y := \text{CORR}(Y)$<br>$s := \text{CORR}(S)$<br>$a := \text{CORR}(A)$<br>$\left(\frac{Z}{(XA^d)^s \cdot B^{e_1ad}}\right)^{e_1^{-1}}$<br>$= B^x := Z^*$ |
| 3   | $Y$ may be from $\mathcal{A}$<br>$b := \text{CORR}(B)$<br>$a := \text{CORR}(A)$<br>$\frac{Z}{(YB^{e_1}S^{e_2})^{ad} \cdot X^{e_1b}}$<br>$= (Y \cdot S^{e_2})^x := Z^*$                                  | No $Y$ involved<br>$b := \text{CORR}(B)$<br>$a := \text{CORR}(A)$<br>$\frac{Z}{(SB^{e_1})^{ad} \cdot X^{e_1b}}$<br>$= S^x := Z^*$                             | $y := \text{CORR}(Y)$<br>$b := \text{CORR}(B)$<br>$a := \text{CORR}(A)$<br>$\left(\frac{Z}{(YB^{e_1}S^{e_2})^{ad} \cdot X^{y+e_1b}}\right)^{e_2^{-1}}$<br>$= S^x := Z^*$ | No $Y$ involved<br>$b := \text{CORR}(B)$<br>$a := \text{CORR}(A)$<br>$\frac{Z}{(SB^{e_1})^{ad} \cdot X^{e_1b}}$<br>$= S^x := Z^*$                         |
| 4   | $S$ may be from $\mathcal{A}$<br>$b := \text{CORR}(B)$<br>$a := \text{CORR}(A)$<br>$\left(\frac{Z}{(YB^{e_1}S^{e_2})^{ad} \cdot X^{e_1b}}\right)^{e_2^{-1}}$<br>$= (S \cdot Y^{e_2^{-1}})^x := Z^*$     | -                                                                                                                                                             | $s := \text{CORR}(S)$<br>$b := \text{CORR}(B)$<br>$a := \text{CORR}(A)$<br>$\frac{Z}{(YB^{e_1}S^{e_2})^{ad} \cdot X^{e_1b+e_2s}}$<br>$= Y^x := Z^*$                      | -                                                                                                                                                         |
| 5   | $Y$ may be from $\mathcal{A}$<br>$S$ may be from $\mathcal{A}$<br>$x := \text{CORR}(X)$<br>$\left(\frac{Z}{(YB^{e_1}S^{e_2})^x}\right)^{d^{-1}}$<br>$= ((YS^{e_2}) \cdot B^{e_1})^a := Z^*$             | No $Y$ involved<br>$S$ may be from $\mathcal{A}$<br>$x := \text{CORR}(X)$<br>$\left(\frac{Z}{(B^{e_1}S)^x}\right)^{d^{-1}}$<br>$= (S \cdot B^{e_1})^a := Z^*$ | $y := \text{CORR}(Y)$<br>$s := \text{CORR}(S)$<br>$X$ may be from $\mathcal{A}$<br>$\left(\frac{Z}{(XA^d)^{y+se_2}}\right)^{e_1^{-1}}$<br>$= (X \cdot A^d)^b := Z^*$     | No $Y$ involved<br>$s := \text{CORR}(S)$<br>$X$ may be from $\mathcal{A}$<br>$\left(\frac{Z}{(XA^d)^s}\right)^{e_1^{-1}}$<br>$= (X \cdot A^d)^b := Z^*$   |
| 6   | $S$ may be from $\mathcal{A}$<br>$b := \text{CORR}(B)$<br>$x := \text{CORR}(X)$<br>$\left(\frac{Z}{(YB^{e_1}S^{e_2})^x \cdot A^{dbe_1}}\right)^{d^{-1}e_2^{-1}}$<br>$= (S \cdot Y^{e_2^{-1}})^a := Z^*$ | -                                                                                                                                                             | $s := \text{CORR}(S)$<br>$b := \text{CORR}(B)$<br>$X$ may be from $\mathcal{A}$<br>$\frac{Z}{(XA^d)^{be_1+se_2}}$<br>$= (X \cdot A^d)^y := Z^*$                          | -                                                                                                                                                         |

-  $t = 1, 2, 3, 4, 5$ , and  $6$  correspond to the LT-SS, Eph-LT, Eph-SS, Eph-Eph, LT-LT, and LT-Eph attacks, respectively.

-  $d := H_0(A, X) = \text{CH}(X)$ , and  $e_1 := H_1(B, Y, S) = \text{CH}(YS^{e_2})$  for the full mode and  $e_1 := H_3(B, S) = \text{CH}(S)$  for the reduced mode.

-  $e_2 := H_2(B, Y, S) = \text{CH}(Y)$  in  $\mathcal{B}_1$ , while in  $\mathcal{B}_2$ ,  $e_2 := H_2(B, Y, S) = (\text{CH}(S))^{-1}$ . Therefore, in cases  $t = 4$  or  $6$ ,  $(S \cdot Y^{e_2^{-1}})^u = (S \cdot Y^{\text{CH}(S)})^u$  (where  $u = x$  or  $a$ ).

**QueryKDF** into two sub-events

$$\Pr[\mathbf{QueryKDF}] = \Pr\left[\mathbf{QueryKDF}_{\neg(\text{LT-Eph}, \text{Eph-Eph})}\right] + \Pr\left[\mathbf{QueryKDF}_{(\text{LT-Eph}, \text{Eph-Eph})}\right],$$

where  $\mathbf{QueryKDF}_{\neg(\text{LT-Eph}, \text{Eph-Eph})}$  is the event that **QueryKDF** happens and the invoked **cleanHMQV** outputs  $d \in \{1, 2, 3, 5\}$  (excluding the LT-Eph and Eph-Eph cases), and  $\mathbf{QueryKDF}_{(\text{LT-Eph}, \text{Eph-Eph})}$  is the event that **QueryKDF** happens and **cleanHMQV** outputs  $d \in \{4, 6\}$  (only considering the LT-Eph and Eph-Eph cases).

Figures 22 and 23 present reductions  $\mathcal{B}_1$  and  $\mathcal{B}_2$  for bounding the probabilities of events  $\mathbf{QueryKDF}_{\neg(\text{LT-Eph}, \text{Eph-Eph})}$  and  $\mathbf{QueryKDF}_{(\text{LT-Eph}, \text{Eph-Eph})}$ , respectively. Figure 22 shows the common part of  $\mathcal{B}_1$  and  $\mathcal{B}_2$ , and Figure 23 shows how they differ in simulating the random oracles and extracting solutions. More specifically,  $\mathcal{B}_1$  and  $\mathcal{B}_2$  rely on Table 2 to compute the **CorrCRGapCDH** solution based on the correct XHMQV DH shared secret  $Z$  in the query to  $\text{KDF}_f$  or  $\text{KDF}_r$ .

Since both  $\mathcal{B}_1$  and  $\mathcal{B}_2$  have complex structures, in the following we explain them step by step.

- First, similar to  $\mathcal{B}_{\text{dh}}$  in Figure 19,  $\mathcal{B}_1$  and  $\mathcal{B}_2$  simulate all public keys via properly distributing their indexes and maintaining the mapping between public keys and their indexes. The simulation of all oracles that are not presented in Figures 22 and 23 is the same as  $\mathbf{G}_6$  (cf. Figure 20). By the definition of **CorrCRGapCDH**, the reductions are given access to **CORR**, **DDH**, and **CH** oracles. The first two are the same as in Figure 19, and the last one is used to simulate hash values in computing XHMQV shared secrets.
- Random oracles  $H_0, H_1, H_2$ , and  $H_3$  are simulated via lazy sampling, but with random responses sampled from **CH**. By the definition of the **CorrCRGapCDH** assumption (cf. Figure 5), the responses from **CH** are distributed uniformly in  $Z_p$ , and repeated queries to **CH** return independent random values. The code can be found in Figure 23. For reduction  $\mathcal{B}_2$ , since it uses the inverse of responses from **CH** to simulate  $H_2$ , we need to add  $\frac{Q_h}{p}$  to the final bound of  $\mathcal{B}_2$  to account for the probability that the inverse does not exist in some query.

Finally, we prove that if  $\mathcal{A}$  makes a query to  $\text{KDF}_f$  or  $\text{KDF}_r$  that makes **cleanHMQV** output a non- $\perp$  value, then either  $\mathcal{B}_1$  or  $\mathcal{B}_2$  can extract a valid **CorrCRGapCDH** solution (i.e., either the event  $\mathbf{QueryKDF}_{\neg(\text{LT-Eph}, \text{Eph-Eph})}$  or event  $\mathbf{QueryKDF}_{(\text{LT-Eph}, \text{Eph-Eph})}$  happens, respectively).

Let  $(Z, A, X, B, S, Y, \sigma)$  be the query that makes **cleanHMQV** output a non- $\perp$  value. Since we assumed  $Z^* \neq \perp$ , there must be a session  $\pi$  such that **cleanHMQV** finds  $\pi$ 's **sid** matches  $(A, X, B, S, Y)$ . Hence, for convenience, we further let  $A, B, \text{ssid}, i$ , and  $j$  be the initiator party, responder party, index of the responder party's semi-static keys, and session indexes of  $\pi$ , respectively.  $\pi$  is either  $\pi_i^A$  or  $\pi_j^B$ , depending on whether  $\pi$  is an initiator or responder session. Let  $Z^*$  be the output of **cleanHMQV** $(A, X, B, S, Y, Z)$  and  $t \in [6]$  be the case index computed during the execution of **cleanHMQV**. We consider the following cases:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Reduction</b> <math>(\mathcal{B}_1, \mathcal{B}_2)^{\text{DDH, CORR, CH}} \left( \begin{pmatrix} (U_i)_{1 \leq i \leq N_{\text{LT}}}, \\ (S_{i,j})_{(i,j) \in [N_{\text{LT}}] \times [n_{\text{SS}}]}, \\ (R_i)_{1 \leq i \leq N_{\text{SS}}} \end{pmatrix} \right)</math></p> <hr/> <pre> 01 <math>\beta \xleftarrow{\\$} \{0, 1\}</math>, <math>\text{cnt} := 0</math>, <math>Z^* \xleftarrow{\\$} \mathbb{G}</math> 02 <b>for</b> <math>P \in [N_{\text{LT}}]</math> 03   <math>(\text{isk}[P], \text{ipk}[P]) := (\perp, U_P)</math> 04   <b>for</b> <math>\text{ssid} \in [n_{\text{SS}}]</math> 05     <math>S := S_{P, \text{ssid}}</math> // <math>((P-1) \cdot n_{\text{SS}} + \text{ssid})</math>-th <math>S</math> 06     <math>(\sigma, q, h) \xleftarrow{\\$} \text{SIMsign}(U_P, S)</math> 07     <math>\mathcal{L}_{\text{RO}} := \mathcal{L}_{\text{RO}} \cup \{(q, h)\}</math> 08     <math>\text{sssk}[P, \text{ssid}] := \perp</math> 09     <math>\text{sspk}[P, \text{ssid}] := (S, \sigma)</math> 10 <math>\beta^* \leftarrow \mathcal{A}^O(\text{ipk}, \text{sspk})</math> 11 // Output a solution if <b>QueryKDF</b> never occurs 12 <b>return</b> <math>Z^*</math> </pre> <p><b>Oracle</b> <math>\text{DER}(P, i, \text{ssid}, m)</math></p> <pre> 13 <math>\pi_i^P := \Pi[P, i]</math> 14 <b>if</b> <math>\pi_i^P = \perp \vee \pi_i^P.\text{SK} \neq \perp</math>: <b>return</b> <math>\perp</math> 15 <math>B := P</math>, <math>(b, B) := (\text{isk}[P], \text{ipk}[P])</math> 16 <math>(S, \sigma) := \text{sspk}[P, \text{ssid}]</math> 17 <math>(\perp, Y) := \pi.\text{state}</math> 18 <math>(A, X) := m</math>, <math>A := \text{ipk}[A]</math> 19 <math>K := \text{PatchSK}(A, X, B, S, Y, \sigma)</math> 20 <math>\pi.(\text{pid}, \text{SK}) := (A, K)</math> 21 <math>\pi.\text{sid} := (A, X, B, S, Y, \sigma)</math> 22 <math>\Pi[P, i] := \pi_i^P</math>; <b>return</b> <math>\epsilon</math> </pre> <p><b>Oracle</b> <math>\text{CORLT}(P)</math></p> <pre> 23 <math>\text{corLT}[P] := \text{true}</math> 24 <b>return</b> <math>\text{CORR}(P)</math> </pre> <p><b>Oracle</b> <math>\text{CORSS}(P, \text{ssid})</math></p> <pre> 25 <math>\text{corSS}[\text{sspk}[P, \text{ssid}]] := \text{true}</math> 26 // <math>S_i</math> is the <math>\text{ssid}</math>-th semi-static pk of <math>P</math> 27 <math>i := (N_{\text{LT}} + (P-1) \cdot n_{\text{SS}} + \text{ssid})</math> 28 <b>return</b> <math>\text{CORR}(i)</math> </pre> | <p><b>Oracle</b> <math>\text{REVRAND}(P, i)</math></p> <pre> 29 <b>if</b> <math>\pi_i^P.\text{rand} \neq \perp</math> 30   <math>\text{revRand}[\pi_i^P] := \text{true}</math> 31   <math>k := \pi_i^P.\text{rand}</math> 32   <b>return</b> <math>\text{CORR}(N_{\text{LT}} + N_{\text{SS}} + k)</math> 33 <b>return</b> <math>\perp</math> </pre> <p><b>Oracle</b> <math>\text{SEND}(P, i, \text{ssid}, m, \text{role}, \text{type})</math></p> <pre> 34 ... // Setup variables of <math>\pi</math> 35 <math>\text{cnt} := \text{cnt} + 1</math> 36 <math>\pi.\text{rand} := \text{cnt}</math> 37 <b>Case</b> <math>\text{role} = \text{R}</math> // InitR 38   <math>B := P</math>, <math>B := \text{ipk}[P]</math> 39   <math>(S, \sigma) := \text{sspk}[P, \text{ssid}]</math> 40   <b>if</b> <math>\text{type} = \text{full}</math>: <math>Y := R_{\text{cnt}}</math> 41   <b>else</b> <math>Y := \perp</math> 42   <math>m' := m_{\text{bundle}} := (B, (S, \sigma), Y)</math> 43   ... // Update variables of <math>\pi</math> 44 <b>Case</b> <math>\text{role} = \text{I}</math>: // InitI 45   <math>(B, (S, \sigma), Y) := m_{\text{bundle}}</math> 46   <math>A := P</math>, <math>A := \text{ipk}[P]</math> 47   <math>B := \text{ipk}[B]</math> 48   <b>if</b> <math>\text{Verify}(B, \sigma, S) \neq \text{true}</math> 49     <b>return</b> <math>\perp</math> 50   <b>else</b> <math>\pi.\text{sspk} := (S, \sigma)</math> 51   <b>if</b> <math>\neg \text{corLT}[B] \wedge (S, \sigma) \notin \text{sspk}[B, \cdot]</math> 52     <b>abort</b> and <b>output</b> <math>Z^*</math> 53   <b>if</b> <math>Y \neq \perp</math>: <math>\pi.\text{type} := \text{full}</math> 54   <b>else</b> <math>\pi.\text{type} := \text{reduced}</math> 55   <math>X := R_{\text{cnt}}</math>, <math>m' := (A, X)</math> 56   <math>K := \text{PatchSK}(A, X, B, S, Y, \sigma)</math> 57   ... // Update variables of <math>\pi</math> 58   <b>return</b> <math>m'</math> </pre> <p><b>Procedure</b> <math>\text{PatchSK}(A, X, B, S, Y, \sigma)</math></p> <pre> 59 <b>if</b> <math>\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] = \perp</math> 60   <math>K \xleftarrow{\\$} \text{SK}</math> 61   <math>\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma] := K</math> 62 <b>return</b> <math>\mathcal{L}_{\text{SK}}[A, X, B, S, Y, \sigma]</math> </pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Fig. 22.** Reductions  $\mathcal{B}_1$  and  $\mathcal{B}_2$  for bounding the probabilities of events  $\text{QueryKDF}_{\neg(\text{LT-Eph, Eph-Eph})}$  and  $\text{QueryKDF}_{(\text{LT-Eph, Eph-Eph})}$  in  $\mathbf{G}_6$ , respectively. The simulation of the presented oracles is the same for both reductions, and the different part is shown in Figure 23.

- $\mathcal{B}_1$  with  $t = 1$ : In this case,  $A$  and  $S$  are uncorrupted (remember that we used the unforgeability of the signature scheme to exclude that  $\mathcal{A}$  uses malicious  $S$ ). By Table 2 and simulation of  $H_0$  in Figure 23, we have  $Z^* = (R \cdot A^d)^s$  where  $d = \text{CH}(R)$  for some  $R$  (either  $X$  or a self-generated value, cf. Lines 51 to 62). Although we also need to get the secret exponents of  $B$  and  $X$  (if  $\pi$  is an initiator session) via querying  $\text{CORR}$ , it does not affect the freshness of

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Oracle <math>H_0(A, X)</math></b></p> <pre> 01 if <math>\mathcal{L}_{H_0}[A, X] = \perp</math> 02   <math>h \leftarrow \text{CH}(X)</math> 03   <math>\mathcal{L}_{H_0}[A, X] := h</math> 04 return <math>\mathcal{L}_{H_0}[A, X]</math> </pre> <p><b>Oracle <math>H_1(B, Y, S)</math></b></p> <pre> 05 if <math>\mathcal{L}_{H_1}[B, Y, S] = \perp</math> 06   <math>e_2 := H_2(B, Y, S)</math> 07   <math>h \leftarrow \text{CH}(Y S^{e_2})</math> 08   <math>\mathcal{L}_{H_1}[B, Y, S] := h</math> 09 return <math>\mathcal{L}_{H_1}[B, Y, S]</math> </pre> <p><b>Oracle <math>H_2(B, Y, S)</math> // For <math>\mathcal{B}_1</math></b></p> <pre> 10 if <math>\mathcal{L}_{H_2}[B, Y, S] = \perp</math> 11   <math>h \leftarrow \text{CH}(Y)</math> 12   <math>\mathcal{L}_{H_2}[B, Y, S] := h</math> 13 return <math>\mathcal{L}_{H_2}[B, Y, S]</math> </pre> <p><b>Oracle <math>H_3(B, S)</math></b></p> <pre> 14 if <math>\mathcal{L}_{H_3}[B, S] = \perp</math> 15   <math>h \leftarrow \text{CH}(S)</math> 16   <math>\mathcal{L}_{H_3}[B, S] := h</math> 17 return <math>\mathcal{L}_{H_3}[B, S]</math> </pre> <p><b>Oracle <math>\text{KDF}_t(Z, A, X, B, S, Y, \sigma)</math></b></p> <pre> 18 <math>d := H_0(A, X)</math>, <math>e_1 := H_1(B, Y, S)</math> 19 <math>e_2 := H_2(B, Y, S)</math> 20 if <math>\text{DDH}(X A^d, Y B^{e_1} S^{e_2}, Z)</math> 21   <math>Z^* := \text{cleanHMQV}(A, X, B, S, Y, Z)</math> 22   if <math>Z^* \neq \perp</math> 23     abort and output <math>Z^*</math> 24 ... // The rest is the same as Figure 20 </pre> <p><b>Oracle <math>\text{KDF}_t(Z, A, X, B, S, \sigma)</math></b></p> <pre> 25 <math>d := H_0(A, X)</math>, <math>e_1 := H_3(B, S)</math> 26 if <math>\text{DDH}(X A^d, B^{e_1} S, Z)</math> 27   <math>Z^* := \text{cleanHMQV}(A, X, B, S, \perp, Z)</math> 28   if <math>Z^* \neq \perp</math> 29     abort and output <math>Z^*</math> 30 return <math>\text{PatchSK}(A, X, B, S, \perp, \sigma)</math> 31 ... // The rest is the same as Figure 20 </pre> | <p><b>Oracle <math>H_2(B, Y, S)</math> // For <math>\mathcal{B}_2</math></b></p> <pre> 32 if <math>\mathcal{L}_{H_2}[B, Y, S] = \perp</math> 33   <math>h \leftarrow \text{CH}(S)</math> 34   <math>\mathcal{L}_{H_2}[B, Y, S] := h^{-1} \bmod p</math> 35 return <math>\mathcal{L}_{H_2}[B, Y, S]</math> </pre> <p><b>Procedure <math>\text{cleanHMQV}(A, X, B, S, Y, Z)</math></b></p> <pre> 36 <math>t := 0</math> 37 Find <math>\pi</math> such that <math>\pi.\text{sid} = (A, X, B, S, Y, \cdot)</math> 38 if <math>\pi = \perp</math>: return <math>\perp</math> // No corresponding sessions 39 Find <math>A</math> s.t. <math>\text{ipk}[A] = A</math> 40 Find <math>B</math> s.t. <math>\text{ipk}[B] = B</math> 41 Find <math>\text{ssid}</math> s.t. <math>\text{sspk}[B, \text{ssid}] = (S, \cdot)</math> 42 Find <math>i</math> s.t. <math>x = \Pi[A, i].\text{rand}</math> and <math>X = g^x</math> 43 Find <math>j</math> s.t. <math>y = \Pi[B, j].\text{rand}</math> and <math>Y = g^y</math> 44 Compute <math>t</math>, <math>\text{role}</math>, and <math>\text{type}</math> as in Figure 21 45 if <math>t = 0</math>: return <math>\perp</math> 46 if <math>t \notin \{1, 2, 3, 5\}</math>: return <math>\perp</math> // <math>\mathcal{B}_1</math> 47 if <math>t \notin \{4, 6\}</math>: return <math>\perp</math> // <math>\mathcal{B}_2</math> 48 <math>\text{type} := \pi.\text{type}</math> // Full or reduced mode 49 Compute <math>Z^*</math> as in Table 2    based on the variables <math>\text{role}</math>, <math>\text{type}</math> and <math>t</math> 50 // Convert <math>Z^*</math> into a <math>\text{CorrCRGapCDH}</math> solution    if it is a pure DH shared secret 51 <b>Case <math>t = 1 \wedge \text{type} = \text{reduced}</math>:</b> // <math>A^s</math> 52   // Construct <math>(RA^{\text{Ch}(R)})^s</math> 53   <math>r \xleftarrow{\\$} \mathbb{Z}_p</math>, <math>Z^* := S^r (Z^*)^{\text{Ch}(g^r)}</math> 54 <b>Case <math>t = 2 \wedge \text{role} = \text{R}</math>:</b> // <math>B^x</math> 55   // Construct <math>(RB^{\text{Ch}(R)})^x</math> 56   <math>r \xleftarrow{\\$} \mathbb{Z}_p</math>, <math>Z^* := X^r (Z^*)^{\text{Ch}(g^r)}</math> 57 <b>Case <math>t = 3 \wedge (\text{role} = \text{R} \vee \text{type} = \text{reduced})</math></b> 58   // Based on <math>S^x</math>, construct <math>(RS^{\text{Ch}(R)})^x</math> 59   <math>r \xleftarrow{\\$} \mathbb{Z}_p</math>, <math>Z^* := X^r (Z^*)^{\text{Ch}(g^r)}</math> 60 <b>Case <math>t = 4 \wedge \text{role} = \text{R}</math></b> // <math>Y^x</math> 61   // Construct <math>(RY^{\text{Ch}(R)})^x</math> 62   <math>r \xleftarrow{\\$} \mathbb{Z}_p</math>, <math>Z^* := X^r (Z^*)^{\text{Ch}(g^r)}</math> 63 return <math>Z^*</math> </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Fig. 23.** Reductions  $\mathcal{B}_1$  and  $\mathcal{B}_2$  for bounding the transition from  $\mathbf{G}_5$  to  $\mathbf{G}_6$ . Omitted oracles are the same as in  $\mathbf{G}_5$ .

$A$  and  $S$  since they are both uncorrupted. Therefore,  $\mathcal{B}_1$  can output  $Z^*$  as a valid **CorrCRGapCDH** solution.

- $\mathcal{B}_1$  with  $t = 2$ : Both  $X$  and  $B$  are uncorrupted, and  $Z^* = (YS^{e_2} \cdot B^{e_1})^x$  or  $(R \cdot B^{e_1})^x$ , depending on whether  $\pi$  is a full initiator session. If so, then  $Z^* = (YS^{e_2} \cdot B^{e_1})^x = (YS^{e_2} \cdot B^{\text{Ch}(YS^{e_2})})^x$ ; otherwise,  $Z^* = (R \cdot B^{e_1})^x = (R \cdot B^{\text{Ch}(R)})^x$  for some  $R$ . Although  $\mathcal{B}_1$  may also need to corrupt  $Y, S$ , and  $A$ , since it does not affect the freshness of  $X$  and  $B$ ,  $\mathcal{B}_1$  can output  $Z^*$  as a valid solution in either case.
- $\mathcal{B}_1$  with  $t = 3$  and  $t = 5$ : These two cases are similar to prior cases.  $Z^*$  always has the form of  $(R \cdot S^{\text{Ch}(R)})^x$  (for  $t = 3$ ) or  $(R \cdot A^{\text{Ch}(R)})^b$  (for  $t = 5$ ) for some  $R$ , where  $R$  is either  $Y$  ( $t = 3$  and initiator session),  $YS^{e_2}$  ( $t = 5$  and full initiator session),  $S$  ( $t = 5$  and reduced initiator session),  $X$  ( $t = 5$  and responder session), or some self-generated value (other cases).
- $\mathcal{B}_2$  with  $t = 4$ : In this case, both  $X$  and  $Y$  are uncorrupted. If  $\pi$  is an initiator session, then  $Z^* = (S \cdot Y^{e_2^{-1}})^x = (S \cdot Y^{\text{Ch}(S)})^x$  (since  $\mathcal{B}_2$  simulates  $e_2 := H_2(B, Y, S) \leftarrow (\text{Ch}(S))^{-1}$ ). If  $\pi$  is a responder session, then by Table 2 and Line 62,  $Z^* = (R \cdot Y^{\text{Ch}(R)})^x$  for some  $R$ . Hence, in either case,  $\mathcal{B}_2$  can output  $Z^*$  as a valid solution.
- $\mathcal{B}_2$  with  $t = 6$ : In this case, both  $A$  and  $Y$  are uncorrupted. By Table 2,  $Z^* = (S \cdot Y^{e_2^{-1}})^a$  if  $\pi$  is an initiator session, and  $Z^* = (X \cdot A^d)^y$  if  $\pi$  is a responder session. For the latter case, we have  $Z^* = (X \cdot A^{\text{Ch}(X)})^y$ . For the former case, which is similar to the case  $t = 4$ , we have  $Z^* = (S \cdot Y^{e_2^{-1}})^a = Z^* = (S \cdot Y^{\text{Ch}(S)})^a$ . Hence, in either case,  $\mathcal{B}_2$  can output  $Z^*$  as a valid solution.

Each reduction makes at most  $Q_h$  queries to  $\text{CH}$  to simulate  $H_0, H_1, H_2$ , and  $H_3$ , and at most one additional query  $\text{CH}(g^r)$  during extraction (cf. Lines 51 to 62). The extraction cases are mutually exclusive, and the reduction terminates after extracting a solution. Thus, each reduction makes at most  $Q_h + 1$  challenge queries in total.

Therefore, either  $\mathcal{B}_1$  or  $\mathcal{B}_2$  outputs a valid **CorrCRGapCDH** solution if **QueryKDF** occurs. Hence, we have

$$\begin{aligned} \Pr[\mathbf{QueryKDF}] &\leq \text{Adv}_{N_{\text{LT}}+N_{\text{SS}}+N_{\text{ses}}, Q_{\text{LT}}+Q_{\text{SS}}+Q_{\text{Eph}}+3, Q_{\text{kdf}}, Q_h+1}^{\text{CorrCRGapCDH}}(\mathcal{B}_1) \\ &\quad + \text{Adv}_{N_{\text{LT}}+N_{\text{SS}}+N_{\text{ses}}, Q_{\text{LT}}+Q_{\text{SS}}+Q_{\text{Eph}}+3, Q_{\text{kdf}}, Q_h+1}^{\text{CorrCRGapCDH}}(\mathcal{B}_2) + \frac{Q_h}{p} \end{aligned}$$

We can combine  $\mathcal{B}_1$  and  $\mathcal{B}_2$  into a single reduction  $\mathcal{B}_{\text{dh}}$  where it samples a random bit to decide which reduction to run. Hence, we have

$$\Pr[\mathbf{QueryKDF}] \leq 2\text{Adv}_{N_{\text{LT}}+N_{\text{SS}}+N_{\text{ses}}, Q_{\text{LT}}+Q_{\text{SS}}+Q_{\text{Eph}}+3, Q_{\text{kdf}}, Q_h+1}^{\text{CorrCRGapCDH}}(\mathcal{B}_{\text{dh}}) + \frac{Q_h}{p}$$

Finally, the remaining analysis of the winning probability of  $\mathcal{A}$  in  $\mathbf{G}_6$  is also the same as in the proof of **X3DH**. Hence, we have  $\Pr[\mathbf{G}_6^{\mathcal{A}} = 1] = \frac{1}{2}$  and

$$\left| \Pr[\text{BAKE}_{\text{XHMqv}}^{\mathcal{A}}(\mathcal{A}) = 1] - \frac{1}{2} \right|$$

$$\leq \text{Adv}_{N_{\text{LT}}, N_{\text{SS}}, Q_{\text{LT}}, Q_{\text{kdf}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{F}) + 2\text{Adv}_{N_{\text{LT}}+N_{\text{SS}}+N_{\text{ses}}, Q_{\text{LT}}+Q_{\text{SS}}+Q_{\text{Eph}}+3, Q_{\text{kdf}}, Q_{\text{h}}+1}^{\text{CorrCRGapCDH}}(\mathcal{B}_{\text{dh}}) \\ + N_{\text{LT}} \cdot \delta_{\text{sim}} + \frac{(N_{\text{LT}} + N_{\text{SS}} + N_{\text{ses}})^2 + 2Q_{\text{h}}}{2p},$$

which concludes the proof of Theorem 2.

## D Omitted proofs in Section 4

### D.1 Fixed-size set sampler Lemma

Inspired by [5], we present the following Lemma 10 for the fixed-size set sampler. This lemma can be viewed as a variant of [5, Theorem 3.3] adapted to the setting where the output consists of two challenges. For readability, we follow the notations in [5] and adapt them as needed.

**Lemma 10.** *Let  $C \subseteq [n]$  be a subset of size  $0 < c \leq n-2$ , and let  $i^*, j^* \in [n] \setminus C$  be two distinct indices. Let  $t \in \{c, \dots, n-2\}$ . Then the probability*

$$\hat{\mathbf{P}}_{n,c,T}(C, i^*, j^*) := \Pr\left[C \subseteq T \wedge i^*, j^* \notin T \mid T \xleftarrow{\$} \mathcal{P}_t([n])\right] = \frac{\binom{n-c-2}{t-c}}{\binom{n}{t}}, \quad (4)$$

*is independent of the particular choice of  $C$ ,  $i^*$ , and  $j^*$ , where  $T \xleftarrow{\$} \mathcal{P}_t([n])$  denotes that  $T$  is chosen uniformly at random from all subsets of  $[n]$  of size  $t$ .*

*Moreover, the above probability is maximized over all  $t \in \{c, \dots, n-2\}$  by  $t = t^* := \lceil \frac{cn-2}{c+2} \rceil$ . For this choice of  $t$ , we define the corresponding probability as  $\hat{\mathbf{P}}^*(n, c)$ , and we have*

$$\hat{\mathbf{P}}^*(n, c) \geq \frac{1}{e^2} \left( \frac{2}{c+2} \right)^2. \quad (5)$$

PROOF. For a uniformly sampled set  $T$  of size  $t$ , the event requires all  $c$  elements of  $C$  to be included and both  $i^*, j^*$  to be excluded. Thus exactly  $\binom{n-c-2}{t-c}$  of the  $\binom{n}{t}$  sets satisfy the event, proving Equation (4) and its independence of the particular choices of  $C$ ,  $i^*$ , and  $j^*$ . For fixed  $n$  and  $c$ , we therefore denote this probability by  $f(t)$ , and analyze its maximum over  $t \in \{c, \dots, n-2\}$ .

If  $c = n-2$ , then  $t = n-2$  is the only admissible choice and hence trivially maximizes  $f(t)$ . Now assume  $0 < c < n-2$ . For  $c \leq t < n-2$ , we have

$$\frac{f(t+1)}{f(t)} = \frac{(n-2-t)(t+1)}{(n-t)(t-c+1)} \geq 1 \iff t \leq \frac{cn-2}{c+2}.$$

Since  $c-1 < (cn-2)/(c+2) \leq n-2$ , the integer  $t^* = \lceil (cn-2)/(c+2) \rceil$  lies in  $\{c, \dots, n-2\}$  and maximizes  $f(t)$ . The same formula remains valid for  $c = n-2$ , in which case it gives  $t^* = n-2$ .

To lower-bound this maximum, sample a set  $U \subseteq [n]$  by including each index independently with probability  $p = c/(c+2)$ . Then

$$\Pr[C \subseteq U \wedge i^*, j^* \notin U] = p^c(1-p)^2.$$

Conditioned on  $|U| = t$ , the set  $U$  is uniform among all subsets of size  $t$ . With  $f(t) = 0$  outside  $\{c, \dots, n-2\}$ , it follows that

$$p^c(1-p)^2 = \sum_{t=0}^n \Pr[|U| = t] \cdot f(t) \leq \sum_{t=0}^n \Pr[|U| = t] \cdot f(t^*) \leq f(t^*).$$

Finally,  $(1 + 2/c)^c \leq e^2$  gives

$$\hat{\mathbf{P}}^*(n, c) = f(t^*) \geq \left(\frac{c}{c+2}\right)^c \left(\frac{2}{c+2}\right)^2 \geq \frac{1}{e^2} \left(\frac{2}{c+2}\right)^2.$$

□

## D.2 Proof of Lemma 1

Let  $\mathcal{A}$  be an adversary against the  $\text{CorrCRGapCDH}$  assumption. Consider the sequence of games  $\mathbf{G}_0$ - $\mathbf{G}_3$  shown in Figure 24.

| Game $\mathbf{G}_0$ - $\mathbf{G}_3$                                                                                             | Oracle $\text{CORR}(A_i)(i \in [n])$                                                 |
|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 01 <b>for</b> $i \in [n]$ : $a_i \xleftarrow{\$} \mathbb{Z}_p$ ; $A_i := g^{a_i}$                                                | 14 <b>if</b> $i \notin T$ // $\mathbf{G}_3$                                          |
| 02 $i' \xleftarrow{\$} [n]$                                                                                                      | 15 $\text{BAD}_2 := \text{true}$ ; <b>abort</b> // $\mathbf{G}_3$                    |
| 03 $T \xleftarrow{\$} \mathcal{P}_t([n])$                                                                                        | 16 $\mathcal{S}_{\text{corr}} := \mathcal{S}_{\text{corr}} \cup \{i\}$               |
| 04 $i' \xleftarrow{\$} [n] \setminus T$                                                                                          | 17 <b>return</b> $a_i$                                                               |
| 05 <b>for</b> $j \in [n] \setminus T$                                                                                            | <b>Oracle CH</b> ( $R_k \in \mathbb{G}$ )                                            |
| 06 $r_j \xleftarrow{\$} \mathbb{Z}_p$ ; $A_j := A_{i'} \cdot g^{r_j}$                                                            | 18 $h_k \xleftarrow{\$} \mathbb{Z}_p$ // $\mathbf{G}_0$                              |
| 07 $Z \xleftarrow{\$} \mathcal{A}^{\text{DDH}, \text{CORR}, \text{CH}}(A_1, \dots, A_n)$                                         | 19 $a_k, b_k \xleftarrow{\$} \mathbb{Z}_p$ // $\mathbf{G}_1$ - $\mathbf{G}_3$        |
| 08 <b>if</b> $\exists(i^*, j^*, k^*) \in ([n] \setminus \mathcal{S}_{\text{corr}})^2 \times [Q_{\text{CH}}] \wedge i^* \neq j^*$ | 20 $R'_k := R_k \cdot A_{i'}^{a_k} \cdot g^{b_k}$ // $\mathbf{G}_1$ - $\mathbf{G}_3$ |
| s.t. $Z = (A_{i^*}^{b_{k^*}} \cdot R_{k^*})^{a_{j^*}}$                                                                           | 21 <b>if</b> $f(R'_k) = 0$ // $\mathbf{G}_2$ - $\mathbf{G}_3$                        |
| 09 <b>if</b> $i^* \in T$ <b>or</b> $j^* \in T$                                                                                   | 22 $\text{BAD}_1 := \text{true}$ ; <b>abort</b> // $\mathbf{G}_2$ - $\mathbf{G}_3$   |
| 10 $\text{BAD}_2 := \text{true}$ ; <b>abort</b>                                                                                  | 23 $h_k := (f(R'_k) + a_k) \bmod p$ // $\mathbf{G}_1$ - $\mathbf{G}_3$               |
| 11 <b>return</b> 1                                                                                                               | 24 <b>return</b> $h_k$                                                               |
| 12 <b>else return</b> 0                                                                                                          |                                                                                      |
| <b>Oracle DDH</b> ( $X_\ell, Y_\ell, Z_\ell$ )                                                                                   |                                                                                      |
| 13 <b>return</b> $[Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)}]$                                                                     |                                                                                      |

Fig. 24. Games  $\mathbf{G}_0$ - $\mathbf{G}_3$  in the proof of Lemma 1.

**Game  $\mathbf{G}_0$ :** This is the attack game against the  $\text{CorrCRGapCDH}$  assumption in the standard model.

$$\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] = \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, Q_{\text{CH}}}^{\text{CorrCRGapCDH}}(\mathcal{A}).$$

**Game  $\mathbf{G}_1$ :** We choose a random index  $i'$  at the beginning of the game and use the  $A_{i'}$  to reprogram the queries to the challenge oracle  $\text{CH}$ .  $A_{i'}$  will be replaced

by a challenge later. For each CH query, instead of returning a random  $h_k$ , we first choose random values  $a_k, b_k \xleftarrow{\$} \mathbb{Z}_p$  and get a randomized  $R'_k := R_k \cdot A_{i'}^{a_k} \cdot g^{b_k}$ . Then we use the function  $f$  to compute  $(f(R'_k) + a_k) \bmod p$  and return it as the corresponding output.

Since  $a_k$  is uniformly random in  $\mathbb{Z}_p$ , the value  $f(R'_k) + a_k$  is uniformly random in  $\mathbb{Z}_p$ , and thus the distribution of  $h_k$  in this game is the same as in  $\mathbf{G}_0$ . Therefore, the distributions in games  $\mathbf{G}_0$  and  $\mathbf{G}_1$  are identical, and we have

$$\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1].$$

**Game  $\mathbf{G}_2$ :** In this game, if  $\mathcal{A}$  queries the challenge oracle CH with a  $R_k$  such that the function value of the randomized  $R'_k$  is zero (i.e.  $f(R'_k) = 0$ ), then we define an event  $\text{BAD}_1$  occurs and abort the game.

Since  $b_k$  is uniformly random and independent of either  $a_k$  or  $R_k$ ,  $R'_k$  is distributed uniformly at random. Therefore, for each query  $\text{CH}(R_k)$ , the probability that  $f(R'_k) = 0$  is at most  $|f^{-1}(0)|/p$ , and we have

$$\left| \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}_1] \leq Q_{\text{CH}} \cdot \frac{|f^{-1}(0)|}{p}.$$

**Game  $\mathbf{G}_3$ :** Similar to the proof of Lemma 11, here we use the fixed-size set sampling to guess  $\mathcal{A}$ 's corruption set with a higher probability than the naive random guessing of  $i^*$  and  $j^*$  (which will lead to a loss of  $O(n^2)$  instead of  $O(Q_{\text{CORR}}^2)$ ). More precisely, we first uniformly sample a random subset  $T \subseteq [n]$  of size  $t = \lceil (n \cdot Q_{\text{CORR}} - 2) / (Q_{\text{CORR}} + 2) \rceil$ . Instead of choosing a random index  $i'$  from  $[n]$ , we choose it from  $[n] \setminus T$ . Since  $T$  is sampled uniformly at random, the distribution of  $i'$  remains uniform over  $[n]$ . Then, for  $j \in [n] \setminus T$ , we use self-randomization to generate  $A_j := A_{i'} \cdot g^{r_j}$ . These modifications do not change  $\mathcal{A}$ 's view.

Then we define the event  $\text{BAD}_2$  as the game aborts if  $\mathcal{A}$  corrupts any index not in  $T$ , or if  $\mathcal{A}$ 's target index  $i^*, j^* \notin [n] \setminus T$ . Since the distributions of  $\mathbf{G}_3$  and  $\mathbf{G}_2$  are the same if  $\text{BAD}_2$  does not occur, we have

$$\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] \geq \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1 \wedge \neg \text{BAD}_2] = \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1 \wedge \neg \text{BAD}_2].$$

By Lemma 10, the probability of  $\neg \text{BAD}_2$  is essentially  $\hat{\mathbf{P}}_{n,c,T}(\mathcal{S}_{\text{CORR}}, i^*, j^*)$  for  $t = \lceil (n \cdot Q_{\text{CORR}} - 2) / (Q_{\text{CORR}} + 2) \rceil$ , and  $c = Q_{\text{CORR}}$ , which is independent of  $\mathcal{S}_{\text{CORR}}$ ,  $i^*$ , and  $j^*$ , and thus the behaviour of  $\mathcal{A}$ . Therefore, we have

$$\begin{aligned} \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] &\geq \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1 \wedge \neg \text{BAD}_2] \\ &= \Pr[\neg \text{BAD}_2] \cdot \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] \\ &\geq \frac{1}{e^2} \left( \frac{2}{Q_{\text{CORR}} + 2} \right)^2 \cdot \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1]. \end{aligned}$$

Now we are ready to bound the winning probability of  $\mathcal{A}$  in game  $\mathbf{G}_3$  by the probability of breaking the  $f$ -GapCSqDH assumption. We construct a reduction

$\mathcal{B}$  as in Figure 25. Here we additionally add the bad event  $\text{BAD}_{\text{COLL}}$  to make sure all the  $A_i$ 's are distinct.  $\text{BAD}_{\text{COLL}}$  can only happen with probability  $n^2/p$ .

After  $\mathcal{A}$  outputs  $Z$ ,  $\mathcal{B}$  uses its own oracle DDH to check, for all the  $R_k$  queried to the CH oracles and all the  $i, j \in [n] \setminus T$ , if there exists a tuple  $(i^*, j^*, k^*)$  such that

$$Z = \text{CDH}((A_{i^*})^{h_{k^*}} \cdot R_{k^*}, A_{j^*}) = \text{CDH}((X \cdot g^{r_{i^*}})^{f(R'_{k^*})+a_{k^*}} \cdot R_{k^*}, X \cdot g^{r_{j^*}}).$$

If such tuple exists, then  $\mathcal{B}$  computes  $Z^* := \frac{Z \cdot (X \cdot g^{r_{j^*}})^{b_{k^*}-r_{i^*} \cdot h_{k^*}}}{(R'_{k^*} \cdot X^{f(R'_{k^*})})^{r_{j^*}}}$  and returns  $(R'_{k^*}, Z^*)$ . We show that  $(R'_{k^*}, Z^*)$  is a valid  $f$ -GapCSqDH response:

By the definition of  $R'_{k^*}$  (cf. Line 15), we have  $R_{k^*} = R'_{k^*} \cdot X^{-a_{k^*}} \cdot g^{-b_{k^*}}$ . Hence,

$$\begin{aligned} Z &= \left( (X \cdot g^{r_{i^*}})^{f(R'_{k^*})+a_{k^*}} \cdot R_{k^*} \right)^{x+r_{j^*}} \\ &= \left( (X \cdot g^{r_{i^*}})^{f(R'_{k^*})+a_{k^*}} \cdot R'_{k^*} \cdot X^{-a_{k^*}} \cdot g^{-b_{k^*}} \right)^{x+r_{j^*}} \\ &= \left( (g^{r_{i^*}})^{f(R'_{k^*})+a_{k^*}} \cdot R'_{k^*} \cdot X^{f(R'_{k^*})} \cdot g^{-b_{k^*}} \right)^{x+r_{j^*}} \\ &= \left( g^{r_{i^*}(f(R'_{k^*})+a_{k^*})-b_{k^*}} \cdot R'_{k^*} \cdot X^{f(R'_{k^*})} \right)^{x+r_{j^*}} \\ &= \left( R'_{k^*} \cdot X^{f(R'_{k^*})} \right)^{x+r_{j^*}} \cdot \left( g^{r_{i^*}h_{k^*}-b_{k^*}} \right)^{x+r_{j^*}}. \end{aligned}$$

Therefore, we have

$$\left( R'_{k^*} \cdot X^{f(R'_{k^*})} \right)^{x+r_{j^*}} = Z / g^{(r_{i^*}h_{k^*}-b_{k^*}) \cdot (x+r_{j^*})} = Z \cdot (X \cdot g^{r_{j^*}})^{b_{k^*}-r_{i^*} \cdot h_{k^*}},$$

and thus

$$Z^* = \frac{Z \cdot (X \cdot g^{r_{j^*}})^{b_{k^*}-r_{i^*} \cdot h_{k^*}}}{(R'_{k^*} \cdot X^{f(R'_{k^*})})^{r_{j^*}}}$$

is the valid response to break  $f$ -GapCSqDH.

If  $\text{BAD}_{\text{COLL}}$  does not happen,  $\mathcal{B}$  simulates  $\mathcal{A}$ 's view in  $\mathbf{G}_3$ . Moreover,  $\mathcal{B}$  breaks the  $f$ -GapCSqDH assumption if  $\mathcal{A}$  wins in  $\mathbf{G}_3$ , and we have

$$\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] \leq \text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{B}) + \Pr[\text{BAD}_{\text{COLL}}] \leq \text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{B}) + \frac{n^2}{p}.$$

Combining the above inequalities, we obtain

$$\begin{aligned} \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, Q_{\text{CH}}}^{\text{CorrCRGapCDH}}(\mathcal{A}) &\leq e^2 \cdot \frac{(Q_{\text{CORR}} + 2)^2}{4} \cdot (\text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{B}) + \frac{n^2}{p}) + Q_{\text{CH}} \cdot \frac{|f^{-1}(0)|}{p} \\ &= e^2 \cdot \frac{(Q_{\text{CORR}} + 2)^2}{4} \cdot \text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{B}) \end{aligned}$$

| Reduction $\mathcal{B}^{\text{DDH}}(X)$                                                                   | Oracle $\text{CH}(R_k \in \mathbb{G})$          |
|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| 01 $T \xleftarrow{\$} \mathcal{P}_t([n])$                                                                 | 14 $a_k, b_k \xleftarrow{\$} \mathbb{Z}_p$      |
| 02 <b>for</b> $i \in T$ : $a_i \xleftarrow{\$} \mathbb{Z}_p$ ; $A_i := g^{a_i}$                           | 15 $R'_k := R_k \cdot X^{a_k} \cdot g^{b_k}$    |
| 03 $i' \xleftarrow{\$} [n] \setminus T$ ; $A_{i'} := X$                                                   | 16 <b>if</b> $f(R'_k) = 0$                      |
| 04 <b>for</b> $j \in [n] \setminus T$ : $r_j \xleftarrow{\$} \mathbb{Z}_p$ ; $A_j := X \cdot g^{r_j}$     | 17 $\text{BAD}_1 := \text{true}$ ; <b>abort</b> |
| 05 <b>if</b> $\exists (i, j) \in [n]$ s.t. $i \neq j \wedge A_i = A_j$                                    | 18 $h_k := (f(R'_k) + a_k) \bmod p$             |
| 06 $\text{BAD}_{\text{COLL}} := \text{true}$ ; <b>abort</b>                                               | 19 <b>return</b> $h_k$                          |
| 07 $Z \xleftarrow{\$} \mathcal{A}^{\text{DDH}, \text{CORR}, \text{CH}}(A_1, \dots, A_n)$                  |                                                 |
| 08 <b>for</b> $(i, j, k) \in ([n] \setminus T)^2 \times [Q_{\text{CH}}]$                                  |                                                 |
| 09 $Z_{i,j,k} := Z \cdot (X \cdot g^{r_j})^{b_k - r_i \cdot h_k}$                                         |                                                 |
| 10 <b>if</b> $\exists (i^*, j^*, k^*) \in ([n] \setminus T)^2 \times [Q_{\text{CH}}] \wedge i^* \neq j^*$ |                                                 |
| s.t. $\text{DDH}((R'_{k^*} \cdot X^{f(R'_{k^*})}), X \cdot g^{r_{j^*}}, Z_{i^*, j^*, k^*}) = 1$           |                                                 |
| 11 $Z^* := Z_{i^*, j^*, k^*} \cdot (R'_{k^*} \cdot X^{f(R'_{k^*})})^{-r_{j^*}}$                           |                                                 |
| 12 <b>return</b> $(R'_{k^*}, Z^*)$                                                                        |                                                 |
| 13 <b>else abort</b>                                                                                      |                                                 |

**Fig. 25.** Reduction  $\mathcal{B}$  against the  $f$ -GapCSqDH assumption. All the other oracles are the same as in  $\mathbf{G}_3$ .

$$\begin{aligned}
& + \frac{e^2 \cdot (Q_{\text{CORR}} + 2)^2 \cdot n^2 + 4Q_{\text{CH}} \cdot |f^{-1}(0)|}{4p} \\
& \leq e^2 \cdot \frac{(Q_{\text{CORR}} + 2)^2}{4} \cdot \text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{B}) + \frac{O(n^4 + Q_{\text{CH}})}{p}.
\end{aligned}$$

### D.3 Proof of Lemma 2

This proof is in the ABROM (cf. Definition 3). Let  $\mathcal{A}$  be an adversary against the  $f$ -GapCSqDH assumption. We define  $\mathbf{G}_0$ - $\mathbf{G}_4$  in Figure 26.

| Game $\mathbf{G}_0$ - $\mathbf{G}_4$                                                                               | Oracle $\text{BRO}(\vec{\rho})$                                                                              | Oracle $\text{BRO}^{-1}(\beta)$                                                                               |
|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 01 $\Pi \xleftarrow{\$} \text{Inj}(\mathbb{A}, \mathbb{B})$                                                        | 15 $R := \prod_i U_i^{\rho_i}$                                                                               | 29 $\alpha := \Pi^{-1}(\beta)$ // $\mathbf{G}_0$                                                              |
| 02 $x \xleftarrow{\$} \mathbb{Z}_p$                                                                                | 16 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                                                                 | 30 <b>if</b> $\exists \alpha$ s.t. $(\alpha, \beta) \in \text{LIST}_{\Pi}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$ |
| 03 $X := g^x$                                                                                                      | 17 $\alpha := \varphi(R)$                                                                                    | 31 <b>return</b> $\alpha$ // $\mathbf{G}_1$ - $\mathbf{G}_4$                                                  |
| 04 $\mathbf{U} := (g, X)$                                                                                          | 18 $\beta := \Pi(\alpha)$ // $\mathbf{G}_0$                                                                  | 32 $\alpha \xleftarrow{\$} \mathbb{A} \setminus \mathcal{S}_{\mathbb{A}}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$  |
| 05 $(R^*, Z) \xleftarrow{\$} \mathcal{A}^{\text{BRO}, \text{BRO}^{-1}, \text{DDH}}(X)$                             | 19 <b>if</b> $\exists \beta$ s.t. $(\alpha, \beta) \in \text{LIST}_{\Pi}$                                    | 33 $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha\}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$  |
| 06 $\alpha^* := \varphi(R^*)$                                                                                      | 20 <b>return</b> $\beta$ // $\mathbf{G}_1$ - $\mathbf{G}_4$                                                  | 34 $\mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta\}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$   |
| 07 $\beta^* := \Pi(\alpha^*)$ // $\mathbf{G}_0$                                                                    | 21 $\beta \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$  | 35 $\text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha, \beta)\}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$       |
| 08 <b>if</b> $\alpha^* \notin \mathcal{S}_{\mathbb{A}}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$                         | 22 $\mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta\}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$  | 36 <b>if</b> $\alpha \in \varphi(\mathbb{G})$ // $\mathbf{G}_1$ - $\mathbf{G}_4$                              |
| 09 $\beta^* \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$      | 23 $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha\}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$ | 37 $(R, R^{-1}) := \varphi^{-1}(\alpha)$ // $\mathbf{G}_0$ - $\mathbf{G}_2$                                   |
| 10 $\text{BAD}_1 \leftarrow \text{true}$ ; <b>abort</b> // $\mathbf{G}_2$ - $\mathbf{G}_4$                         | 24 $\text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha, \beta)\}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$      | 38 $s, t \xleftarrow{\$} \mathbb{Z}_p$ // $\mathbf{G}_3$ - $\mathbf{G}_4$                                     |
| 11 <b>else fetch</b> $\beta^*$ s.t. $(\alpha^*, \beta^*) \in \text{LIST}_{\Pi}$ // $\mathbf{G}_1$ - $\mathbf{G}_4$ | 25 <b>let</b> $(s, t)$ be $R = g^s X^t$ // $\mathbf{G}_4$                                                    | 39 $R := g^s X^t$ // $\mathbf{G}_3$ - $\mathbf{G}_4$                                                          |
| 12 $c := \Psi(\beta^*)$ // $\mathbf{G}_1$ - $\mathbf{G}_4$                                                         | 26 <b>if</b> $t = -\Psi(\beta)$ // $\mathbf{G}_4$                                                            | 40 $\alpha \leftarrow \varphi(R)$ // $\mathbf{G}_3$ - $\mathbf{G}_4$                                          |
| 13 <b>return</b> $\llbracket c \neq 0 \wedge Z = (R^* \cdot X^c)^* \rrbracket$                                     | 27 $\text{BAD}_2 \leftarrow \text{true}$ ; <b>abort</b> // $\mathbf{G}_4$                                    | 41 <b>if</b> $t = -\Psi(\beta)$ // $\mathbf{G}_4$                                                             |
|                                                                                                                    | 28 <b>return</b> $\beta$                                                                                     | 42 $\text{BAD}_2 \leftarrow \text{true}$ ; <b>abort</b> // $\mathbf{G}_4$                                     |
|                                                                                                                    |                                                                                                              | 43 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                                                                  |
| Oracle $\text{DDH}(X_{\ell}, Y_{\ell}, Z_{\ell})$                                                                  |                                                                                                              | 44 <b>return</b> $\alpha$                                                                                     |
| 14 <b>return</b> $\llbracket Z_{\ell} = Y_{\ell}^{\text{DLOG}_g(X_{\ell})} \rrbracket$                             |                                                                                                              |                                                                                                               |

**Fig. 26.** Games  $\mathbf{G}_0$ - $\mathbf{G}_4$  in the proof of Lemma 2. Here  $\mathbb{A} := \{0, 1\}^L$ ,  $\mathbb{B} := [2^L]$ , and  $\text{Inj}(\mathbb{A}, \mathbb{B})$  denotes all the bijective functions mapping from  $\mathbb{A}$  to  $\mathbb{B}$ .

**Game  $\mathbf{G}_0$ :** This is the security game against the  $f$ -GapCSqDH assumption in the ABROM. Without loss of generality, we assume for each query to BRO, the input is the representation vector  $\vec{\rho}$  corresponding to the group element  $R = \prod_i U_i^{\vec{\rho}_i}$ , where vector  $\mathbf{U} := (U_1, \dots, U_n)$  keeps track of all the group elements that have been involved, and it will be kept updated.

$$\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] = \text{Adv}_{Q_{\text{DDH}}}^{f\text{-GapCSqDH}}(\mathcal{A}).$$

**Game  $\mathbf{G}_1$ :** In this game, we use lazy sampling for the bijective random permutation  $\Pi$  by reprogramming oracles BRO and  $\text{BRO}^{-1}$ . Specifically speaking, instead of choosing uniformly at random a function  $\Pi$  at the beginning of the game, we generate random responses for each new query to  $\Pi$  and  $\Pi^{-1}$ , and record all the input-output pairs in the list  $\text{LIST}_{\Pi}$  in order to keep the consistency. Moreover, we use the sets  $\mathcal{S}_{\mathbb{A}}$  and  $\mathcal{S}_{\mathbb{B}}$  to avoid collisions so that  $\Pi$  acts as a bijective function. Since the distribution of the function  $\Pi$  in  $\mathbf{G}_1$  and  $\mathbf{G}_0$  are identical, we have

$$\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1].$$

**Game  $\mathbf{G}_2$ :** In this game, during the finalization phase, we define an event  $\text{BAD}_1$  occurs if  $\alpha^* \notin \mathcal{S}_{\mathbb{A}}$ , and abort the game. This corresponds to the case where the adversary has queried neither BRO on the representation vector  $\vec{\rho}$  of  $\varphi^{-1}(\alpha)$  nor  $\text{BRO}^{-1}$  on  $\Pi(\alpha)$ . If  $\text{BAD}_1$  happens, then  $\beta^*$  is chosen uniformly at random and hidden from  $\mathcal{A}$ 's view. This means that  $c = \Psi(\beta)$  is uniformly random and independent of  $X, R^*, Z$ . Therefore, the equation  $Z = (R^* \cdot X^c)^x$  holds with probability at most  $\text{MAX}_{\Psi}/p$ .  $\mathbf{G}_1$  and  $\mathbf{G}_2$  are distributed exactly the same if  $\text{BAD}_1$  does not happen, and  $\mathcal{A}$  wins in  $\mathbf{G}_2$  with probability at most  $\text{MAX}_{\Psi}/p$  when  $\text{BAD}_1$  happens. Hence, we have

$$\left| \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}_1] \leq \frac{\text{MAX}_{\Psi}}{p}.$$

**Game  $\mathbf{G}_3$ :** In this game, we modify the way of getting  $\varphi^{-1}(\Pi^{-1}(\beta))$  in the oracle  $\text{BRO}^{-1}$ . If  $\alpha = \Pi^{-1}(\beta)$  is in the range  $\varphi(\mathbb{G})$ , then we choose two random  $s, t \xleftarrow{\$} \mathbb{Z}_p$  and program the preimage  $R := \varphi^{-1}(\alpha)$  as the group element  $g^s X^t$ . Since here  $\alpha$  is chosen uniformly at random from  $\mathbb{A} \setminus \mathcal{S}_{\mathbb{A}}$  and  $\varphi$  is a semi-injective function, the distribution of its (semi-)preimage should also be uniformly at random. Therefore, our modification does not change the distribution of the game, and we have

$$\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1].$$

**Game  $\mathbf{G}_4$ :** Our modification in  $\mathbf{G}_3$  makes sure that all the group elements  $R$  appended to the list  $\mathbf{U}$  in the  $\text{BRO}^{-1}$  can be represented as  $g^s X^t$ . Moreover, since the list  $\mathbf{U}$  is initialized as  $(g, X)$  and each new  $R$  appended to  $\mathbf{U}$  in the BRO has the representation  $\prod_i U_i^{\vec{\rho}_i}$ , all the elements in  $\mathbf{U}$  should have the form  $R = g^s X^t$  for some  $s, t \in \mathbb{Z}_p$ .

In this game, we define the event  $\text{BAD}_2$  occurs if for any query to the BRO or the  $\text{BRO}^{-1}$ , the corresponding representation vector  $(s, t)$  of  $R = g^s X^t$  satisfies

$t = -\Psi(\beta)$ , and abort the game. Since  $\beta$  is chosen uniformly at random in the oracle BRO, the probability that BAD<sub>2</sub> happens in each query to BRO is at most  $\text{MAX}_\Psi/p$ . On the other hand, since  $t$  is chosen uniformly at random in the oracle  $\text{BRO}^{-1}$ , BAD<sub>2</sub> happens in each query to  $\text{BRO}^{-1}$  is at most  $1/p$ . Therefore, we have

$$\left| \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}_2] \leq q \cdot \frac{1 + \text{MAX}_\Psi}{p}.$$

| Reduction $\mathcal{B}^{\text{DDH}}(X)$                                                | Oracle $\text{BRO}(\tilde{\rho})$                                       | Oracle $\text{BRO}^{-1}(\beta)$                                          |
|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------------|
| 01 $\mathbf{U} := (g, X)$                                                              | 13 $R := \prod_i U_i^{\tilde{\rho}_i}$                                  | 26 <b>if</b> $\exists \alpha$ s.t. $(\alpha, \beta) \in \text{LIST}_\Pi$ |
| 02 $(R^*, Z) \xleftarrow{\$} \mathcal{A}^{\text{DDH}, \text{BRO}, \text{BRO}^{-1}}(X)$ | 14 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                            | 27 <b>return</b> $\alpha$                                                |
| 03 $\alpha^* := \varphi(R^*)$                                                          | 15 $\alpha := \varphi(R)$                                               | 28 $\alpha \xleftarrow{\$} \mathbb{A} \setminus \mathcal{S}_\mathbb{A}$  |
| 04 <b>if</b> $\alpha^* \notin \mathcal{S}_\mathbb{A}$                                  | 16 <b>if</b> $\exists \beta$ s.t. $(\alpha, \beta) \in \text{LIST}_\Pi$ | 29 $\mathcal{S}_\mathbb{A} := \mathcal{S}_\mathbb{A} \cup \{\alpha\}$    |
| 05 $\beta^* \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_\mathbb{B}$               | 17 <b>return</b> $\beta$                                                | 30 $\mathcal{S}_\mathbb{B} := \mathcal{S}_\mathbb{B} \cup \{\beta\}$     |
| 06 <b>BAD</b> <sub>1</sub> $\leftarrow$ <b>true</b> ; <b>abort</b>                     | 18 $\beta \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_\mathbb{B}$  | 31 $\text{LIST}_\Pi := \text{LIST}_\Pi \cup \{(\alpha, \beta)\}$         |
| 07 <b>else</b> $\exists \beta^*$ s.t. $(\alpha^*, \beta^*) \in \text{LIST}_\Pi$        | 19 $\mathcal{S}_\mathbb{B} := \mathcal{S}_\mathbb{B} \cup \{\beta\}$    | 32 <b>if</b> $\alpha \in \varphi(\mathbb{G})$                            |
| 08 <b>let</b> $(s^*, t^*)$ <b>be</b> $R^* = g^{s^*} X^{t^*}$                           | 20 $\mathcal{S}_\mathbb{A} := \mathcal{S}_\mathbb{A} \cup \{\alpha\}$   | 33 $(R, R^{-1}) := \varphi^{-1}(\alpha)$                                 |
| 09 $c := \Psi(\beta^*)$                                                                | 21 $\text{LIST}_\Pi := \text{LIST}_\Pi \cup \{(\alpha, \beta)\}$        | 34 $s, t \xleftarrow{\$} \mathbb{Z}_p$                                   |
| 10 $Z' := (Z/X^{s^*})^{1/(t^*+c)}$                                                     | 22 <b>let</b> $(s, t)$ <b>be</b> $R = g^s X^t$                          | 35 $R := g^s X^t$                                                        |
| 11 <b>return</b> $Z'$                                                                  | 23 <b>if</b> $t = -\Psi(\beta)$                                         | 36 $\alpha \leftarrow \varphi(R)$                                        |
| Oracle $\text{DDH}(X_\ell, Y_\ell, Z_\ell)$                                            | 24 <b>BAD</b> <sub>2</sub> $\leftarrow$ <b>true</b> ; <b>abort</b>      | 37 <b>if</b> $t = -\Psi(\beta)$                                          |
| 12 <b>return</b> $[Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)}]$                           | 25 <b>return</b> $\beta$                                                | 38 <b>BAD</b> <sub>2</sub> $\leftarrow$ <b>true</b> ; <b>abort</b>       |
|                                                                                        |                                                                         | 39 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                             |
|                                                                                        |                                                                         | 40 <b>return</b> $\alpha$                                                |

**Fig. 27.** Reduction to the GapSqDH assumption.

Now we construct a reduction  $\mathcal{B}$  as in Figure 27 to bound the winning probability of  $\mathcal{A}$  in  $\mathbf{G}_4$ . With  $\mathcal{A}$ 's output  $(R^*, Z)$ ,  $\mathcal{B}$  first checks whether the event BAD<sub>1</sub> occurs by verifying if  $\alpha^* = \varphi(R^*)$  is contained in the set  $\mathcal{S}_\mathbb{A}$ . If BAD<sub>1</sub> does not occur, then  $R^*$  must be a group element that has been added to the list  $\mathbf{U}$  during the execution of BRO or  $\text{BRO}^{-1}$ , which means that  $\mathcal{B}$  knows its algebraic representation  $(s^*, t^*)$ . In this case,  $\mathcal{B}$  can compute  $Z' := (Z/X^{s^*})^{1/(t^*+\Psi(\beta))}$  and returns  $Z'$ . Here  $t^* + \Psi(\beta) \neq 0$ , since we abort the game if BAD<sub>2</sub> (i.e.,  $t = -\Psi(\beta)$ ) happens. We show that  $Z'$  is a valid solution to the GapSqDH instance:

If  $\mathcal{A}$  wins in  $\mathbf{G}_4$ , then we have  $Z = (R^* \cdot X^{f(R^*)})^x$ , which is equivalent to

$$Z = (g^{s^*} X^{t^*} \cdot X^{\Psi(\beta)})^x = (g^{s^*} X^{t^*+c})^x = X^{s^*} \cdot (X^x)^{t^*+\Psi(\beta)}.$$

Hence,

$$X^x = (Z/X^{s^*})^{1/(t^*+\Psi(\beta))} = Z'.$$

Since  $\mathcal{B}$  simulates  $\mathcal{A}$ 's view in  $\mathbf{G}_4$ , and breaks the GapSqDH assumption when  $\mathcal{A}$  wins in  $\mathbf{G}_4$ , we have

$$\Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] \leq \text{Adv}_{Q^{\text{GapSqDH}}}^{\text{GapSqDH}}(\mathcal{B}).$$

Combining the above inequalities, we have

$$\text{Adv}_{Q^{\text{f-GapCSqDH}}}^{\text{f-GapCSqDH}}(\mathcal{A}) \leq \text{Adv}_{Q^{\text{GapSqDH}}}^{\text{GapSqDH}}(\mathcal{B}) + \frac{q + (q+1) \cdot \text{MAX}_\Psi}{p}.$$

#### D.4 Proof of Lemma 3

Our proof is largely based on the proof of [5, Theorem 3.3], and here we only sketch the important steps. Similar to [5], we show that  $m$ -GapCDH implies CorrGapCDH with a loss of  $O(Q_{\text{CORR}}^2)$  (cf. Lemma 11). Here the  $m$ -GapCDH assumption states that an adversary is given the  $m$ -tuple random elements  $(g^{a_1}, \dots, g^{a_m})$  and oracle access to DDH as defined in Figure 3, and to win the adversary must output an element  $g^{a_i a_j}$  for  $i, j \in [m]$  and  $i \neq j$ . We denote the advantage of  $\mathcal{B}$  breaking the  $m$ -GapCDH assumption as  $\text{Adv}_m^{\text{GapCDH}}(\mathcal{B})$ . By the random self-reducibility, we can show that the GapCDH assumption tightly implies the  $m$ -GapCDH assumption with a loss of 2 [39, Lemma 9]. This completes the proof of Lemma 3.

Hence, the remaining proof is on Lemma 11, and we state it as follows:

**Lemma 11.** *Let  $n$  and  $Q_{\text{CORR}}$  be integers such that  $0 < Q_{\text{CORR}} \leq n - 2$ , and let  $m = \lfloor (2n + 2)/(Q_{\text{CORR}} + 2) \rfloor \in \{2, \dots, n\}$ . For any adversary  $\mathcal{A}$  against the CorrGapCDH assumption as defined in Definition 6, there exists an adversary  $\mathcal{B}$  against the  $m$ -GapCDH assumption such that  $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ , and*

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{CorrGapCDH}}(\mathcal{A}) \leq \frac{e^2 \cdot (Q_{\text{CORR}} + 2)^2}{4} \cdot \text{Adv}_m^{\text{GapCDH}}(\mathcal{B}).$$

PROOF. Let  $\mathcal{A}$  be the adversary against the CorrGapCDH assumption and  $\mathcal{B}$  against the  $m$ -GapCDH assumption. In the reduction,  $\mathcal{B}$  first uniformly samples a random subset  $T \subseteq [n]$  of size  $t = \lceil (n \cdot Q_{\text{CORR}} - 2)/(Q_{\text{CORR}} + 2) \rceil$ . Sampling  $T \xleftarrow{\$} \mathcal{P}_t([n])$  can be done efficiently in  $O(n)$  time using the Fisher–Yates shuffle algorithm [26].

Since  $m = n - t$ ,  $\mathcal{B}$  uses its  $m$  input challenges from the  $m$ -GapCDH assumption to simulate the challenges corresponding to indices  $i \in [n] \setminus T$ . For  $i \in T$ ,  $\mathcal{B}$  can directly simulate the corresponding challenges by sampling  $a_i \xleftarrow{\$} \mathbb{Z}_p$ , and thus it can answer all corruption queries for indices in  $T$ . Then  $\mathcal{B}$  runs  $\mathcal{A}$  on the simulated set of  $n$  challenges, answers all DDH queries by issuing the same query to its own gap oracle, and outputs  $\mathcal{A}$ 's output.

Let  $C \subseteq [n]$  denote the corruption set of  $\mathcal{A}$  with size  $|C| = Q_{\text{CORR}}$ . Let  $i^*, j^* \in [n]$  be the two indices of the Diffie–Hellman challenges output by  $\mathcal{A}$ . Then, when  $\mathcal{A}$  wins, we have  $i^* \neq j^*$  and  $i^*, j^* \in [n] \setminus C$ . Moreover, if  $\mathcal{A}$  never corrupts indices in  $[n] \setminus T$  and  $i^*, j^* \notin T$ , then  $\mathcal{B}$  can use the output of  $\mathcal{A}$  to break the  $m$ -GapCDH assumption with the same success probability. In other words, if  $C \subseteq T$  and  $i^*, j^* \notin T$ , then  $\mathcal{B}$  perfectly simulates  $\mathcal{A}$ 's view and can break  $m$ -GapCDH when  $\mathcal{A}$  wins.

Therefore, by Lemma 10, under this fixed-size set sampling, the probability that  $C \subseteq T$  and  $i^*, j^* \notin T$  is at least  $\frac{1}{e^2} \cdot \left( \frac{2}{Q_{\text{CORR}} + 2} \right)^2$ . Hence, we have

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{CorrGapCDH}}(\mathcal{A}) \leq \frac{e^2 \cdot (Q_{\text{CORR}} + 2)^2}{4} \cdot \text{Adv}_m^{\text{GapCDH}}(\mathcal{B}).$$

□

### D.5 Proof of Lemma 4

We prove this by using the self-reducibility of the CDH problem and give the reduction in Figure 28. The simulator  $\mathcal{B}$  first chooses a random  $r$ , and computes

---

**Reduction  $\mathcal{B}_{Q_{\text{DDH}}}^{\text{DDH}}(X \in \mathbb{G})$**

```

01  $r \xleftarrow{\$} \mathbb{Z}_p$ 
02  $Y := X \cdot g^r$ 
03  $Z \xleftarrow{\$} \mathcal{A}_{Q_{\text{DDH}}}^{\text{DDH}}(X, Y)$ 
04 return  $Z/X^r$ 

```

---

**Fig. 28.** Reduction from the GapCDH assumption to the GapSqDH assumption.

$Y$  as  $X \cdot g^r$ . Then it runs  $\mathcal{A}$  on the input  $(X, Y)$  and answers the DDH queries by using its own DDH oracle. Finally, after receiving  $\mathcal{A}$ 's output  $Z$ , it returns  $Z/X^r$  as the solution of the square CDH instance.

Since  $\mathcal{B}$  perfectly simulates  $\mathcal{A}$ 's view, and  $\mathcal{B}$  wins if  $\mathcal{A}$  wins, we have

$$\text{Adv}_{Q_{\text{DDH}}}^{\text{GapCDH}}(\mathcal{A}) \leq \text{Adv}_{Q_{\text{DDH}}}^{\text{GapSqDH}}(\mathcal{B}).$$

### D.6 Proof of Lemma 5

Let  $\mathcal{A}$  be an adversary against the GapSqDH assumption. We define  $\mathbf{G}_0$ - $\mathbf{G}_1$  as in Figure 29.

| <b>Game <math>\mathbf{G}_0</math>-<math>\mathbf{G}_1</math></b> | <b>Oracle <math>\text{DDH}(X_\ell, Y_\ell, Z_\ell)</math></b>                                                                   |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 01 $x \xleftarrow{\$} \mathbb{Z}_p$                             | 05 <b>return</b> $\llbracket Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)} \rrbracket$ // $\mathbf{G}_0$                              |
| 02 $X := g^x$                                                   | 06 <b>let</b> $X_\ell = g^{a_1} X^{b_1}, Y_\ell = g^{a_2} X^{b_2}, Z_\ell = g^{a_3} X^{b_3}$                                    |
| 03 $Z \xleftarrow{\$} \mathcal{A}(X)$                           | 07 <b>if</b> $b_1 = 0 \wedge b_3 = b_2 \cdot a_1$ : <b>return</b> $\llbracket a_3 = a_1 \cdot a_2 \rrbracket$ // $\mathbf{G}_1$ |
| 04 <b>return</b> $\llbracket Z = g^{x^2} \rrbracket$            | 08 <b>if</b> $b_2 = 0 \wedge b_3 = b_1 \cdot a_2$ : <b>return</b> $\llbracket a_3 = a_1 \cdot a_2 \rrbracket$ // $\mathbf{G}_1$ |
|                                                                 | 09 <b>if</b> $Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)}$ : <b>BAD</b> := <b>true</b> ; <b>abort</b> // $\mathbf{G}_1$             |
|                                                                 | 10 <b>return</b> 0 // $\mathbf{G}_1$                                                                                            |

**Fig. 29.** Games  $\mathbf{G}_0$ - $\mathbf{G}_1$  in the proof of Lemma 5.

**Game  $\mathbf{G}_0$ :** This is game GapSqDH in the AGM. Since our proof is in the AGM, the input of the DDH is modeled algebraically, namely, we assume for each query  $(X_\ell, Y_\ell, Z_\ell)$   $\mathcal{A}$  must provide the corresponding algebraic representation as given in Line 06. Here  $\mathcal{A}$  only gets  $g$  from the public parameters and  $X$  as group element inputs, and thus we assume the representation is wrt to  $(g, X)$ .

$$\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] = \text{Adv}_{Q_{\text{DDH}}}^{\text{GapSqDH}}(\mathcal{A}).$$

**Game  $\mathbf{G}_1$ :** We modify the simulation of the DDH oracle. More precisely, given  $(X_\ell, Y_\ell, Z_\ell)$ , we use the equation  $a_3 = a_1 \cdot a_2$  to check if the input is a CDH tuple, if one of the following two conditions is satisfied:

- $b_1 = 0$  and  $b_3 = b_2 \cdot a_1$ ;
- $b_2 = 0$  and  $b_3 = b_1 \cdot a_2$ ;

We note that

$$\begin{aligned} Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)} &\Leftrightarrow a_3 + x \cdot b_3 = (a_2 + x \cdot b_2)(a_1 + x \cdot b_1) \\ &\Leftrightarrow (b_1 b_2) \cdot x^2 + (b_1 a_2 + b_2 a_1 - b_3) \cdot x + a_1 a_2 - a_3 = 0 \end{aligned} \quad (6)$$

Hence, if one of the conditions holds, then checking  $a_1 a_2 = a_3$  is equivalent to checking  $Z_\ell = Y_\ell^{\text{DLOG}_g(X_\ell)}$ . This modification will not change the distribution.

If none of the aforementioned two conditions holds, we define the event **BAD** occurs if the input  $(X_\ell, Y_\ell, Z_\ell)$  is a CDH tuple and abort the game. Otherwise, we just return 0.

Now we bound the probability of **BAD** by the advantage of an adversary  $\mathcal{B}$  against the DL assumption. For each DDH query, if **BAD** happens, then Equation (6) holds, which is either a quadratic or linear equation of  $x$  because at least one of the coefficients is non-zero when none of the above two conditions holds. Therefore, Equation (6) can be viewed as a function of  $x$  with degree at most 2.

When  $\mathcal{A}$  queries the DDH, the simulator  $\mathcal{B}$  first checks if one of the above two conditions holds, and if so, it returns  $\llbracket a_3 = a_1 \cdot a_2 \rrbracket$ . Otherwise, it checks if the quadratic equation in Equation (6) has a root satisfying  $X = g^x$ . If such a solution exists,  $\mathcal{B}$  returns  $x$  as the solution of the DL instance. Since the existence of the root is equivalent to the occurrence of **BAD**, and the distribution of the game changes only when **BAD** occurs, we have

$$\left| \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}] \leq \text{Adv}^{\text{DL}}(\mathcal{B}).$$

If **BAD** does not happen, we can use the representation from  $\mathcal{A}$  to simulate DDH. Thus, we reduce the winning probability of  $\mathcal{A}$  in game  $\mathbf{G}_1$  to the SqDH adversary  $\mathcal{C}$ 's advantage as in Figure 30.  $\mathcal{C}$  runs  $\mathcal{A}$  with the same input, simulates the DDH oracle as in game  $\mathbf{G}_1$ , and returns the same output of  $\mathcal{A}$ . Therefore, we have

$$\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \leq \text{Adv}^{\text{SqDH}}(\mathcal{C}).$$

| <u>Reduction <math>\mathcal{C}(X)</math></u>       | <u>Oracle <math>\text{DDH}(X_\ell, Y_\ell, Z_\ell)</math></u>                                                 |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 01 $Z \xleftarrow{\$} \mathcal{A}^{\text{DDH}}(X)$ | 03 <b>let</b> $X_\ell = g^{a_1} X^{b_1}, Y_\ell = g^{a_2} X^{b_2}, Z_\ell = g^{a_3} X^{b_3}$                  |
| 02 <b>return</b> $Z$                               | 04 <b>if</b> $b_1 = 0 \wedge b_3 = b_2 \cdot a_1$ : <b>return</b> $\llbracket a_3 = a_1 \cdot a_2 \rrbracket$ |
|                                                    | 05 <b>if</b> $b_2 = 0 \wedge b_3 = b_1 \cdot a_2$ : <b>return</b> $\llbracket a_3 = a_1 \cdot a_2 \rrbracket$ |
|                                                    | 06 <b>return</b> 0                                                                                            |

**Fig. 30.** Reduction  $\mathcal{C}$  to the SqDH assumption.

Combining the above inequalities, we have

$$\text{Adv}_{Q_{\text{DDH}}}^{\text{GapSqDH}}(\mathcal{A}) \leq \text{Adv}^{\text{SqDH}}(\mathcal{C}) + \text{Adv}^{\text{DL}}(\mathcal{B}).$$

## E Omitted proofs in Section 5

### E.1 Security Proof of XEdDSA (Theorem 5)

Let  $\mathcal{A}$  be an adversary against the  $\text{EUF}^c\text{-1CMA-DDH}$  security of the XEdDSA scheme. Consider the sequence of games  $\mathbf{G}_0\text{-}\mathbf{G}_4$  shown in Figure 31.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Game <math>\mathbf{G}_0\text{-}\mathbf{G}_4</math></b><br>01 <b>for</b> $i \in [n]$ : $a_i \xleftarrow{\$} \mathbb{Z}_p$ ; $A_i := g^{a_i}$<br>02 $(i^*, m^*, (R^*, s^*)) \xleftarrow{\$} \mathcal{A}^{\text{DDH}, \text{CORR}, \text{SIGN}, G, H}((A_i)_{1 \leq i \leq n})$<br>03 <b>if</b> $\mathcal{L}_H[R^* \  A_{i^*} \  m^*] = \perp$ : $\text{BAD}_2 := \text{true}$ ; <b>abort</b> // $\mathbf{G}_4$<br>04 $h^* := H(R^* \  A_{i^*} \  m^*)$<br>05 <b>return</b> $\left[ \begin{array}{l} i^* \notin \mathcal{S}_{\text{corr}} \wedge \mathcal{L}_{\text{sig}}[i^*, m^*] = \perp \\ \wedge g^{s^*} = R^* \cdot A_{i^*}^{h^*} \end{array} \right]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  | <b>Oracle <math>\text{CORR}(i \in [n])</math></b><br>23 $\mathcal{S}_{\text{corr}} := \mathcal{S}_{\text{corr}} \cup \{i\}$<br>24 <b>return</b> $a_i$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Oracle <math>\text{SIGN}(i, m)</math></b><br>06 <b>if</b> $\mathcal{L}_{\text{sig}}[i, m] \neq \perp$ : <b>return</b> $\perp$<br>07 $\text{INP} \xleftarrow{\$} \{0, 1\}^{\text{len}}$<br>08 $\mathcal{S}_{\text{INP}} := \mathcal{S}_{\text{INP}} \cup \{\text{INP}\}$<br>09 $r \leftarrow G(a_i \  m \  \text{INP})$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>10 $r \xleftarrow{\$} \mathbb{Z}_p$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>11 <b>if</b> $\mathcal{L}_G[a_i \  m \  \text{INP}] \neq \perp$ <b>or</b> $r \in \mathcal{L}_G$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>12 $\text{BAD}_G^{\text{COLL}} := \text{true}$ ; <b>abort</b> // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>13 $\mathcal{L}_G[a_i \  m \  \text{INP}] := r$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>14 $R := g^r$ // $\mathbf{G}_0\text{-}\mathbf{G}_1$<br>15 $h \leftarrow H(R \  A_i \  m)$ // $\mathbf{G}_0\text{-}\mathbf{G}_1$<br>16 $s := r + h \cdot a_i$ // $\mathbf{G}_0\text{-}\mathbf{G}_1$<br>17 $s \xleftarrow{\$} \mathbb{Z}_p$ , $h \xleftarrow{\$} \mathbb{Z}_p$ , $R := g^s / A_i^h$ // $\mathbf{G}_2\text{-}\mathbf{G}_4$<br>18 <b>if</b> $\mathcal{L}_H[R \  A_i \  m] \neq \perp$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>19 $\text{BAD}_H^{\text{COLL}} := \text{true}$ ; <b>abort</b> // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>20 $\mathcal{L}_H[R \  A_i \  m] := h$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>21 $\sigma := (R, s)$ , $\mathcal{L}_{\text{sig}}[i, m] := \sigma$<br>22 <b>return</b> $\mathcal{L}_{\text{sig}}[i, m]$ |  | <b>Oracle <math>\text{DDH}(X_\ell, Y_\ell, Z_\ell)</math></b><br>25 <b>return</b> $[Z_\ell = X_\ell^{\text{DL}_g(Y_\ell)}]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  | <b>Oracle <math>G(a_i \  m \  \text{INP})</math></b><br>26 <b>if</b> $\text{INP} \in \mathcal{S}_{\text{INP}}$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>27 $\text{BAD}_G^{\text{COLL}} := \text{true}$ // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>28 <b>abort</b> // $\mathbf{G}_1\text{-}\mathbf{G}_4$<br>29 <b>if</b> $\mathcal{L}_G[a_i \  m \  \text{INP}] = \perp$<br>30 $g \xleftarrow{\$} \mathbb{Z}_p$<br>31 $\mathcal{L}_G[a_i \  m \  \text{INP}] := g$<br>32 <b>return</b> $\mathcal{L}_G[a_i \  m \  \text{INP}]$                                                                                                                                                                                |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  | <b>Oracle <math>H(R \  A_i \  m)</math></b><br>33 <b>if</b> $\mathcal{L}_H[R \  A_i \  m] = \perp$<br>34 $u, v \xleftarrow{\$} \mathbb{Z}_p$ // $\mathbf{G}_3\text{-}\mathbf{G}_4$<br>35 $R' := R \cdot A_i^u \cdot g^v$ // $\mathbf{G}_3\text{-}\mathbf{G}_4$<br>36 <b>if</b> $f(R') = 0$ // $\mathbf{G}_3\text{-}\mathbf{G}_4$<br>37 $\text{BAD}_1 := \text{true}$ // $\mathbf{G}_3\text{-}\mathbf{G}_4$<br>38 <b>abort</b> // $\mathbf{G}_3\text{-}\mathbf{G}_4$<br>39 $h := f(R') + u$ // $\mathbf{G}_3\text{-}\mathbf{G}_4$<br>40 $h \xleftarrow{\$} \mathbb{Z}_p$ // $\mathbf{G}_0\text{-}\mathbf{G}_2$<br>41 $\mathcal{L}_H[R \  A_i \  m] := h$<br>42 <b>return</b> $\mathcal{L}_H[R \  A_i \  m]$ |

Fig. 31. Games  $\mathbf{G}_0\text{-}\mathbf{G}_4$  in the proof of Theorem 5.

**Game  $\mathbf{G}_0$ :** This is the XEdDSA security game against  $\text{EUF}^c\text{-1CMA-DDH}$  in the standard model.

$$\text{Adv}_{n, Q_S, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{A}) = \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1].$$

**Game  $\mathbf{G}_1$ :** In this game, for the signing queries  $\text{SIGN}(i, m)$ , instead of computing  $r$  as the hash function value, we directly choose it uniformly at random. If a collision occurs for the corresponding input or output of  $G$ , we define  $\text{BAD}_G^{\text{COLL}}$  to occur and abort the game. Since  $\text{INP}$  is the signing randomness, the probability of  $\text{BAD}_G^{\text{COLL}}$  is negligible. Now  $R := g^r$  is a random group element. In addition, we define  $\text{BAD}_H^{\text{COLL}}$  to occur if  $(R \| A_i \| m)$  has been queried to the random oracle  $H$ , which is also negligible. Since the distribution remains the same when these

two bad events do not happen, we have

$$\begin{aligned} \left| \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] \right| &\leq \Pr[\text{BAD}_G^{\text{COLL}}] + \Pr[\text{BAD}_H^{\text{COLL}}] \\ &\leq \frac{Q_S \cdot Q_G}{p} + \frac{Q_S \cdot Q_G}{2^{\text{len}}} + \frac{Q_S \cdot Q_H}{p}. \end{aligned}$$

**Game  $\mathbf{G}_2$ :** In this game, we change the simulation for the oracle  $\text{SIGN}$  by reprogramming the random oracle  $H$  such that the secret key is no longer needed. For each signing query  $(i, m)$ , we choose both  $s$  and  $h$  uniformly random and use the verification equation to recover  $R$ , then we reprogram the corresponding value  $H(R \| A_i \| m)$  as  $h$ . Since the distribution of  $R$  is still uniformly random and this change will only be noticed if  $\text{BAD}_H^{\text{COLL}}$  happens, which is already defined and aborts the game in  $\mathbf{G}_1$ , we have

$$\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1].$$

**Game  $\mathbf{G}_3$ :** In this game, we use the function  $f$  to reprogram the random oracle  $H$ . Specifically speaking, we choose two random values  $u, v$  for each fresh query, and use them to re-randomize  $R$  as  $R'$ . Then, we abort the game if  $f(R') = 0$ , and define the output as  $f(R') + u$ , which remains uniformly random. Similar to the analysis of game  $\mathbf{G}_2$  for Lemma 1, we have

$$\left| \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}_1] \leq Q_H \cdot \frac{|f^{-1}(0)|}{p}.$$

**Game  $\mathbf{G}_4$ :** In this game, if  $\mathcal{A}$  outputs a  $R^*$  that has never been queried to the random oracle  $H$ , then we define  $\text{BAD}_2$  to occur and abort the game. If  $\text{BAD}_2$  happens, then the value of  $h^*$  should be uniformly random and hidden from  $\mathcal{A}$ , which indicates that the equation  $g^{s^*} = R^* \cdot A_{i^*}^{h^*}$  holds with probability at most  $\frac{1}{p}$ . Since  $\mathcal{A}$ 's view remains the same when  $\text{BAD}_2$  does not happen, we have

$$\left| \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}_2] \leq \frac{1}{p}.$$

Now we reduce the winning probability of  $\mathcal{A}$  in  $\mathbf{G}_4$  to  $\mathcal{B}$ 's advantage against the  $f$ -CorrGapCDL assumption, as shown in Figure 32.

Given the challenge input  $X_1, \dots, X_n$ , the simulator  $\mathcal{B}$  forwards them to  $\mathcal{A}$  as its input. In the simulation of  $\text{SIGN}$ , Line 13 is the only difference from  $\mathbf{G}_4$ , which does not change the distribution, since the event  $\text{BAD}_G^{\text{COLL}}$  already excludes any query to  $G$  that ends with  $\text{INP}$ . When  $\mathcal{A}$  queries the corruption oracle  $\text{CORR}$  or the gap oracle  $\text{DDH}$ ,  $\mathcal{B}$  answers the queries by invoking its own corresponding oracles. After  $\mathcal{A}$  outputs the forgery, since the event  $\text{BAD}_2$  ensures that  $\mathcal{A}$ 's output  $R^*$  must have been queried to the random oracle  $H$  and that  $m^*$  has not been queried to the signing oracle  $\text{SIGN}$ , there must exist an exponent pair  $(u^*, v^*)$  used to re-randomize  $R^*$  when reprogramming  $H$ . Therefore,  $\mathcal{B}$  computes  $R' = R^* \cdot A_{i^*}^{u^*} \cdot g^{v^*}$ , and outputs  $(R', s^* + v^*)$  as the solution instance

for the  $f$ -CorrGapCDL assumption. Note that the case  $f(R') = 0$  has already been excluded by the event  $\text{BAD}_1$  in Line 37. Hence, we have

$$\text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{f\text{-CorrGapCDL}}(\mathcal{B}) \leq \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1].$$

| Reduction $\mathcal{B}^{\text{CORR, DDH}}(X_i)_{1 \leq i \leq n}$                                               | Oracle $\text{SIGN}(i, m)$                                                                        |
|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 01 <b>for</b> $i \in [n]$ : $A_i := X_i$                                                                        | 09 <b>if</b> $\mathcal{L}_{\text{sig}}[i, m] \neq \perp$ : <b>return</b> $\perp$                  |
| 02 $(i^*, m^*, (R^*, s^*)) \xleftarrow{\$} \mathcal{A}^{\text{DDH, CORR, SIGN, G, H}}((A_i)_{1 \leq i \leq n})$ | 10 $\text{INP} \xleftarrow{\$} \{0, 1\}^{\text{len}}$                                             |
| 03 <b>if</b> $\mathcal{L}_H[R^* \  A_{i^*} \  m^*] = \perp$ : $\text{BAD}_2 := \text{true}$ ; <b>abort</b>      | 11 $\mathcal{S}_{\text{INP}} := \mathcal{S}_{\text{INP}} \cup \{\text{INP}\}$                     |
| 04 $h := H(R^* \  A_{i^*} \  m^*)$                                                                              | 12 $r \xleftarrow{\$} \mathbb{Z}_p$                                                               |
| 05 <b>let</b> $u^*, v^*$ be s.t. $h = f(R^* \cdot A_{i^*}^{u^*} \cdot g^{v^*}) + u^*$                           | 13 <b>if</b> $\mathcal{L}_G[\cdot \  m \  \text{INP}] \neq \perp$ <b>or</b> $r \in \mathcal{L}_G$ |
| 06 <b>if</b> $i^* \notin \mathcal{S}_{\text{corr}} \wedge \mathcal{L}_{\text{sig}}[i^*, m^*] = \perp$           | 14 $\text{BAD}_G^{\text{COLL}} := \text{true}$ ; <b>abort</b>                                     |
| 07 <b>return</b> $(R^* \cdot A_{i^*}^{u^*} \cdot g^{v^*}, s^* + v^*)$                                           | 15 $s \xleftarrow{\$} \mathbb{Z}_p, h \xleftarrow{\$} \mathbb{Z}_p, R := g^s / A_i^h$             |
| 08 <b>else abort</b>                                                                                            | 16 <b>if</b> $\mathcal{L}_H[R \  A_i \  m] \neq \perp$                                            |
|                                                                                                                 | 17 $\text{BAD}_H^{\text{COLL}} := \text{true}$ ; <b>abort</b>                                     |
|                                                                                                                 | 18 $\mathcal{L}_H[R \  A_i \  m] := h$                                                            |
|                                                                                                                 | 19 $\sigma := (R, s), \mathcal{L}_{\text{sig}}[i, m] := \sigma$                                   |
|                                                                                                                 | 20 <b>return</b> $\mathcal{L}_{\text{sig}}[i, m]$                                                 |

**Fig. 32.** Reduction to the  $f$ -CorrGapCDL assumption. All other oracles are the same as in  $\mathbf{G}_4$  in Figure 31.

Combining the above inequalities, we have

$$\begin{aligned} \text{Adv}_{n, Q_S, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\text{EUF}^c\text{-1CMA-DDH}}(\mathcal{A}) &\leq \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{f\text{-CorrGapCDL}}(\mathcal{B}) \\ &\quad + \frac{Q_S \cdot (2Q_G + Q_H) + Q_H \cdot |f^{-1}(0)| + 1}{p} + \frac{Q_S \cdot Q_G}{2^{\text{len}}}. \end{aligned}$$

## E.2 Proof of Theorem 6

Before showing the generic bound of  $f$ -CorrGapCDL, we recall the following useful lemma.

**Lemma 12 (Schwartz-Zippel Lemma).** *Let  $p(x_1, \dots, x_n)$  be a non-zero multivariate polynomial over a field  $\mathbb{F}$ .  $p$  is of maximum degree  $d \geq 0$ . Let  $S$  be a finite subset of  $\mathbb{F}$  and  $a_1, \dots, a_n$  be random elements from  $S$ . We have*

$$\Pr[p(a_1, \dots, a_n) = 0] \leq \frac{d}{|S|}.$$

We use a symbolic way to simulate the game  $f\text{-CorrGapCDL}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\mathcal{A}}$  in the EC-GGM as shown in Figure 33. Specifically, we simulate the encoding function  $\tau$  on the fly, by using a list  $\mathcal{L}_\tau$  to maintain the encodings of group elements  $P \in E$  via degree-1 polynomials  $U(\vec{X})$ . The adversary queries the oracle  $\text{ADD}$  to compute linear combinations of group elements.

---

|                                                                                                           |                                                                                                       |
|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| $f\text{-CorrGapCDL}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\mathcal{A}}$ in EC-GGM                        | <b>Oracle</b> CORR( $i$ ) // $i \in [n]$                                                              |
| 01 $\mathcal{L}_\tau := \{(0, 1_E)\}$                                                                     | 24 $\mathcal{C} := \mathcal{C} \cup \{i\}$                                                            |
| 02 $g \xleftarrow{\$} E^*, \mathcal{L}_\tau := \mathcal{L}_\tau \cup \{(-1, g^{-1}), (1, g)\}$            | 25 <b>if</b> $\exists (U(\vec{X}), P), (V(\vec{X}), P') \in \mathcal{L}_\tau$                         |
| 03 <b>for</b> $i \in [n]$                                                                                 | s.t. $U(\vec{X}) \neq V(\vec{X}) \wedge U(\vec{a}) = V(\vec{a})$                                      |
| 04 $a_i \xleftarrow{\$} \mathbb{Z}_p^*$                                                                   | 26 <b>BAD<sub>COLL</sub> := true; abort</b>                                                           |
| 05 $\text{sample } P_i \xleftarrow{\$} E^*$ until $(\cdot, P_i) \notin \mathcal{L}_\tau$                  | 27 Replace $X_i$ in $\vec{X}$ with $a_i$ and get $\vec{X}'$                                           |
| 06 $\mathcal{L}_\tau := \mathcal{L}_\tau \cup \{(-X_i, P_i^{-1}), (X_i, P_i)\}$                           | 28 <b>for</b> $(U(\vec{X}), P) \in \mathcal{L}_\tau$                                                  |
| 07 $\vec{X} := (X_1, \dots, X_n)$                                                                         | 29     Replace $(U(\vec{X}), P)$ with $(U(\vec{X}'), P)$                                              |
| 08 $\vec{a} := (a_1, \dots, a_n)$                                                                         | 30 $\vec{X} := (X_i)_{i \in [n] \setminus \mathcal{C}}$                                               |
| 09 $(R, z(\vec{X})) \xleftarrow{\$} \mathcal{A}^O(P_1, \dots, P_n)$                                       | 31 $\vec{a} := (a_i)_{i \in [n] \setminus \mathcal{C}}$                                               |
| 10 <b>if</b> $(\cdot, R) \notin \mathcal{L}_\tau$ <b>or</b> $(z(\vec{X}), \cdot) \notin \mathcal{L}_\tau$ | 32 <b>return</b> $a_i$                                                                                |
| 11 <b>BAD<sub>G</sub> := true; return 0</b>                                                               | <b>Oracle</b> ADD( $s, P, t, P'$ )                                                                    |
| 12 fetch $(r(\vec{X}), R), (z(\vec{X}), Z) \in \mathcal{L}_\tau$                                          | 33 fetch $(U(\vec{X}), P), (V(\vec{X}), P') \in \mathcal{L}_\tau$                                     |
| 13 <b>if</b> $\exists (U(\vec{X}), P), (V(\vec{X}), P') \in \mathcal{L}_\tau$                             | 34 $W(\vec{X}) := s \cdot U(\vec{X}) + t \cdot V(\vec{X})$                                            |
| s.t. $U(\vec{X}) \neq V(\vec{X}) \wedge U(\vec{a}) = V(\vec{a})$                                          | 35 <b>if</b> $\exists (W(\vec{X}), \hat{P}) \in \mathcal{L}_\tau$                                     |
| 14 <b>BAD<sub>COLL</sub> := true; abort</b>                                                               | 36 <b>return</b> $\hat{P}$                                                                            |
| 15 <b>if</b> $\exists i^* \in ([n] \setminus \mathcal{C})$ s.t. $f(R) \neq 0$                             | 37 sample $\hat{P} \xleftarrow{\$} E^*$ until $(\cdot, \hat{P}) \notin \mathcal{L}_\tau$              |
| $\wedge z(\vec{a}) = r(\vec{a}) + f(R) \cdot a_{i^*}$                                                     | 38 $\mathcal{L}_\tau := \mathcal{L}_\tau \cup \{(W(\vec{X}), \hat{P}), (-W(\vec{X}), \hat{P}^{-1})\}$ |
| 16 <b>return 1</b>                                                                                        | 39 <b>return</b> $\hat{P}$                                                                            |
| 17 <b>else return 0</b>                                                                                   |                                                                                                       |
| <b>Oracle</b> DDH( $P_{\ell_1}, P_{\ell_2}, P_{\ell_3}$ )                                                 |                                                                                                       |
| 18 fetch $(U(\vec{X}), P_{\ell_1}), (V(\vec{X}), P_{\ell_2})$                                             |                                                                                                       |
| <b>and</b> $(W(\vec{X}), P_{\ell_3}) \in \mathcal{L}_\tau$                                                |                                                                                                       |
| 19 <b>if</b> $W(\vec{X}) = U(\vec{X}) \cdot V(\vec{X})$                                                   |                                                                                                       |
| 20 <b>return 1</b>                                                                                        |                                                                                                       |
| 21 <b>if</b> $W(\vec{a}) = U(\vec{a}) \cdot V(\vec{a})$                                                   |                                                                                                       |
| 22 <b>BAD<sub>DDH</sub> := true; abort</b>                                                                |                                                                                                       |
| 23 <b>return 0</b>                                                                                        |                                                                                                       |

---

**Fig. 33.** Game  $f\text{-CorrGapCDL}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\mathcal{A}}$  in the EC-GGM.  $\mathcal{A}$  has access to oracles  $\mathcal{O} := \{\text{DDH}, \text{CORR}, \text{ADD}\}$ .  $E^* := E \setminus \{1_E\}$ , where  $1_E$  is the identity element of  $E$ .

Without loss of generality, we assume that  $\mathcal{A}$  never queries the oracles DDH or ADD on group elements  $P$  that are not contained in the list, namely, there is no  $(\cdot, P) \in \mathcal{L}_\tau$ . This will not affect the collision probability, since in such cases we can simply sample a fresh random constant as the corresponding polynomial.

If  $\text{BAD}_{\text{COLL}}$  does not occur, then for each DDH query, when  $W(\vec{X}) \neq U(\vec{X}) \cdot V(\vec{X})$ , the degree-2 polynomial  $W(\vec{X}) - U(\vec{X}) \cdot V(\vec{X})$  is non-zero. Hence, by using Lemma 12, we have  $\Pr[\text{BAD}_{\text{DDH}} \mid \neg \text{BAD}_{\text{COLL}}] \leq \frac{2 \cdot Q_{\text{DDH}}}{p-1}$ .

Similar to the analysis in [39, Theorem 2], we define  $Q_{\text{ADD}} := q_0 + \dots + q_{Q_{\text{CORR}}}$ , where  $q_j$  is the number of group operation queries between the  $j$ -th and the  $(j+1)$ -th corruption queries, and  $q_{Q_{\text{CORR}}}$  is the number of such queries made after the last corruption query until the end of the game. Let  $\text{BAD}_{\text{COLL}}^j$  denote the event that  $\text{BAD}_{\text{COLL}}$  first occurs on the  $(j+1)$ -th query to CORR (or at the end of the game in Line 14 when  $j = Q_{\text{CORR}}$ ). Then, by using Lemma 12, we have

$$\begin{aligned} \Pr[\text{BAD}_{\text{COLL}}^0] &\leq \binom{2q_0 + 2n + 1}{2} \cdot \frac{1}{p-1} \leq \frac{(2q_0 + 2n + 1)^2}{2(p-1)}, \\ \forall j \in [Q_{\text{CORR}}] : \Pr[\text{BAD}_{\text{COLL}}^j] &\leq \left( \binom{2q_j}{2} + q_j \cdot \left( \sum_{k \in [0, j-1]} 2q_k + 2n + 1 \right) \right) \cdot \frac{1}{p-1} \\ &\leq \frac{q_j \cdot (\sum_{k \in [0, j]} 2q_k + 2n + 1)}{p-1}. \end{aligned}$$

Therefore, we have

$$\Pr[\text{BAD}_{\text{COLL}}] \leq \sum_{j \in [0, Q_{\text{CORR}}]} \Pr[\text{BAD}_{\text{COLL}}^j] \leq \frac{(2Q_{\text{ADD}} + 2n + 1)^2}{p-1}.$$

Now we consider the probability that, if  $\text{BAD}_{\mathbb{G}}$  does not occur,  $\mathcal{A}$  wins the  $f$ -CorrGapCDL game in the EC-GGM. Let the coefficient of  $x_{i^*}$  in the polynomial  $z(\vec{X})$  and  $r(\vec{X})$  be  $z_{i^*} \in \mathbb{Z}_p$  and  $r_{i^*} \in \mathbb{Z}_p$  respectively. Then,

$$\begin{aligned} &\Pr[\exists i^* \in ([n] \setminus \mathcal{C}) : f(R) \neq 0 \wedge z(\vec{X}) = r(\vec{X}) + f(R) \cdot X_{i^*}] \\ &\leq \Pr[\exists i^* \in ([n] \setminus \mathcal{C}) : f(R) \neq 0 \wedge f(R) = z_{i^*} - r_{i^*}] \\ &\leq \Pr_{\vec{a}}[\exists i^* \in ([n] \setminus \mathcal{C}) : f(r(\vec{a})) \neq 0 \wedge f(r(\vec{a})) = z_{i^*} - r_{i^*}] \\ &\leq \frac{(n - Q_{\text{CORR}}) \cdot \text{MAX}_f}{p-1}. \end{aligned}$$

When  $\text{BAD}_{\mathbb{G}}$  occurs,  $\mathcal{A}$  wins the game with probability at most  $\frac{1}{p-1}$ . Therefore, by combining this with the above inequalities, we obtain

$$\begin{aligned} \text{Adv}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}, \text{EC-GGM}}^{f\text{-CorrGapCDL}}(\mathcal{A}) &\leq \Pr[\text{BAD}_{\text{COLL}}] + \Pr[\text{BAD}_{\text{DDH}} \mid \neg \text{BAD}_{\text{COLL}}] + \Pr[\text{BAD}_{\mathbb{G}}] \\ &\quad + \Pr[f\text{-CorrGapCDL}_{n, Q_{\text{CORR}}, Q_{\text{DDH}}}^{\mathcal{A}} \text{ returns } 1 \mid \neg \text{BAD}_{\mathbb{G}}] \\ &\leq \frac{(2Q_{\text{ADD}} + 2n + 1)^2}{p-1} + \frac{2Q_{\text{DDH}}}{p-1} + \frac{1}{p-1} \end{aligned}$$

$$\begin{aligned}
& + \frac{(n - Q_{\text{CORR}}) \cdot \text{MAX}_f}{p - 1} \\
& \leq \frac{(2Q_{\text{ADD}} + 2n + 1)^2 + (n - Q_{\text{CORR}}) \cdot \text{MAX}_f + 2Q_{\text{DDH}} + 1}{p - 1}.
\end{aligned}$$

### E.3 Proof of Lemma 9

This proof is essentially the same as the proof for Lemma 5. Generally speaking, we reduce the probability of BAD to  $\mathcal{B}$ 's advantage against the DL assumption. Through the whole proof, we only have to replace Line 04 in Figure 29 into  $\llbracket Z = x \rrbracket$  and all others remain the same.

### E.4 Security Proof of ECDSA (Theorem 7)

Let  $\mathcal{A}$  be an adversary against the EUF-CMA security of the ECDSA scheme. The attack game in the ABROM is shown in Figure 34 as game  $\mathbf{G}_0$ , and we have

$$\text{Adv}_{Q_S, Q_H}^{\text{EUF-CMA}}(\mathcal{A}) = \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1].$$

Throughout the proof, we simulate  $H$  as a random oracle using lazy sampling. For simplicity, we do not describe this simulation in the games, but it is implicitly assumed that all queries to  $H$  are answered consistently with a lazy-sampled random oracle.

| Game $\mathbf{G}_0$                                                                               | Oracle $\text{BRO}(\vec{p})$                  | Oracle $\text{SIGN}(m_\ell)$                                          |
|---------------------------------------------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------------------------------|
| 01 $\Pi \xleftarrow{\$} \text{Inj}(\mathbb{A}, \mathbb{B})$                                       | 11 $R \leftarrow \prod_i U_i^{\vec{p}_i}$     | 21 $k_\ell \xleftarrow{\$} \mathbb{Z}_p$                              |
| 02 $x \xleftarrow{\$} \mathbb{Z}_p; X := g^x; pk := X; sk := x$                                   | 12 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$  | 22 $R_\ell := g^{k_\ell}$                                             |
| 03 $\mathbf{U} := (g, X)$                                                                         | 13 $\alpha := \varphi(R)$                     | 23 $\mathbf{U} := \mathbf{U} \  R_\ell \  R_\ell^{-1}$                |
| 04 $(m^*, (r^*, s^*)) \xleftarrow{\$} \mathcal{A}^{\text{BRO}, \text{BRO}^{-1}, \text{SIGN}}(pk)$ | 14 $\beta := \Pi(\alpha)$                     | 24 $\alpha_\ell \leftarrow \varphi(R_\ell)$                           |
| 05 <b>if</b> $m^* \in \mathcal{S}_{\text{sig}}$                                                   | 15 <b>return</b> $\beta$                      | 25 $\beta_\ell \leftarrow \Pi(\alpha_\ell)$                           |
| 06 <b>return</b> 0                                                                                | Oracle $\text{BRO}^{-1}(\beta)$               | 26 $r_\ell \leftarrow \Psi(\beta_\ell)$                               |
| 07 $R^* \leftarrow (g^{H(m^*)} \cdot X^{r^*})^{1/s^*}$                                            | 16 $\alpha := \Pi^{-1}(\beta)$                | 27 $s_\ell := k_\ell^{-1}(H(m_\ell) + xr_\ell)$                       |
| 08 $\alpha^* := \varphi(R^*)$                                                                     | 17 <b>if</b> $\alpha \in \varphi(\mathbb{G})$ | 28 $\mathcal{S}_{\text{sig}} := \mathcal{S}_{\text{sig}} \cup m_\ell$ |
| 09 $\beta^* := \Pi(\alpha^*)$                                                                     | 18 $(R, R^{-1}) := \varphi^{-1}(\alpha)$      | 29 <b>return</b> $(r_\ell, s_\ell)$                                   |
| 10 <b>return</b> $\llbracket r^* = \Psi(\beta^*) \rrbracket$                                      | 19 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$  |                                                                       |
|                                                                                                   | 20 <b>return</b> $\alpha$                     |                                                                       |

**Fig. 34.** Game  $\mathbf{G}_0$  in the proof of Theorem 7. Here  $\mathbb{A} := \{0, 1\}^L$ ,  $\mathbb{B} := [2^L]$ , and  $\text{Inj}(\mathbb{A}, \mathbb{B})$  denotes all the bijective functions mapping from  $\mathbb{A}$  to  $\mathbb{B}$ .

Consider the sequence of games  $\mathbf{G}_1$ - $\mathbf{G}_5$  shown in Figure 35.

**Game  $\mathbf{G}_1$ :** As in the proof of Lemma 2, we first lazily sample the bijective random function  $\Pi$ . Since the distributions are not changed, we have

$$\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1].$$

| Game $G_1$ - $G_5$                                                                                                                                                               | Oracle $BRO(\bar{\rho})$                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| 01 $x \xleftarrow{\$} \mathbb{Z}_p$ ; $X := g^x$ ; $pk := X$ ; $sk := x$                                                                                                         | 30 $R \leftarrow \prod_i U_i^{\bar{\rho}_i}$                                                        |
| 02 $\mathbf{U} := (g, X)$                                                                                                                                                        | 31 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                                                        |
| 03 $(m^*, (r^*, s^*)) \xleftarrow{\$} \mathcal{A}^{BRO, BRO^{-1}, SIGN}(pk)$                                                                                                     | 32 $\alpha := \varphi(R)$                                                                           |
| 04 <b>if</b> $m^* \in \mathcal{S}_{sig}$ : <b>return</b> $\perp$                                                                                                                 | 33 <b>if</b> $\exists \beta$ s.t. $(\alpha, \beta) \in \text{LIST}_{\Pi}$                           |
| 05 $R^* \leftarrow (g^{H(m^*)} \cdot X^{r^*})^{1/s^*}$                                                                                                                           | 34 <b>return</b> $\beta$                                                                            |
| 06 $\alpha^* := \varphi(R^*)$                                                                                                                                                    | 35 $\beta \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}$                            |
| 07 <b>if</b> $\alpha^* \notin \mathcal{S}_{\mathbb{A}}$ : $\beta^* \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}$                                                | 36 $\mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta\}$                            |
| 08 $\text{BAD}_1 \leftarrow \text{true}$ ; <b>abort</b>                                                                                                                          | 37 $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha\}$                           |
| 09 <b>fetch</b> $\beta^*$ s.t. $(\alpha^*, \beta^*) \in \text{LIST}_{\Pi}$                                                                                                       | 38 $\text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha, \beta)\}$                                |
| 10 <b>return</b> $\llbracket r^* = \Psi(\beta^*) \rrbracket$                                                                                                                     | 39 <b>let</b> $(v, w)$ <b>be</b> $R = g^v X^w$ // $G_5$                                             |
|                                                                                                                                                                                  | 40 <b>if</b> $w = -\Psi(\beta)$ // $G_5$                                                            |
|                                                                                                                                                                                  | 41 $\text{BAD}_2 \leftarrow \text{true}$ ; <b>abort</b> // $G_5$                                    |
|                                                                                                                                                                                  | 42 <b>return</b> $\beta$                                                                            |
| <b>Oracle</b> $SIGN(m_{\ell})$                                                                                                                                                   | <b>Oracle</b> $BRO^{-1}(\beta)$                                                                     |
| 11 $k_{\ell} \xleftarrow{\$} \mathbb{Z}_p$ , $R_{\ell} := g^{k_{\ell}}$ , $\alpha_{\ell} \leftarrow \varphi(R_{\ell})$ // $G_1$ - $G_3$                                          | 43 <b>if</b> $\exists \alpha$ s.t. $(\alpha, \beta) \in \text{LIST}_{\Pi}$ : <b>return</b> $\alpha$ |
| 12 <b>if</b> $\exists \beta$ s.t. $(\alpha_{\ell}, \beta) \in \text{LIST}_{\Pi}$ // $G_1$ - $G_3$                                                                                | 44 $\alpha \xleftarrow{\$} \mathbb{A} \setminus \mathcal{S}_{\mathbb{A}}$                           |
| 13 $\text{BAD}_{\text{COLL}0.5} := \text{true}$ ; <b>abort</b> // $G_3$                                                                                                          | 45 $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha\}$                           |
| 14 $\beta_{\ell} := \beta$ // $G_1$ - $G_2$                                                                                                                                      | 46 $\mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta\}$                            |
| 15 <b>else</b> $\beta_{\ell} \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}$ // $G_1$ - $G_3$                                                                     | 47 $\text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha, \beta)\}$                                |
| 16 $\mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta_{\ell}\}$ , $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha_{\ell}\}$ // $G_1$ - $G_3$ | 48 <b>if</b> $\alpha \in \varphi(\mathbb{G})$                                                       |
| 17 $\text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha_{\ell}, \beta_{\ell})\}$ // $G_1$ - $G_3$                                                                              | 49 $(R, R^{-1}) := \varphi^{-1}(\alpha)$ // $G_1$ - $G_4$                                           |
| 18 $s_{\ell} \xleftarrow{\$} \mathbb{Z}_p$ // $G_4$ - $G_5$                                                                                                                      | 50 $v, w \xleftarrow{\$} \mathbb{Z}_p$ // $G_5$                                                     |
| 19 $\beta_{\ell} \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}$ , $\mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta_{\ell}\}$ // $G_4$ - $G_5$  | 51 $R := g^v X^w$ // $G_5$                                                                          |
| 20 $r_{\ell} \leftarrow \Psi(\beta_{\ell})$                                                                                                                                      | 52 $\alpha \leftarrow \varphi(R)$ // $G_5$                                                          |
| 21 $R_{\ell} := (g^{H(m_{\ell})} \cdot X^{r_{\ell}})^{1/s_{\ell}}$ // $G_4$ - $G_5$                                                                                              | 53 <b>if</b> $w = -\Psi(\beta)$ // $G_5$                                                            |
| 22 $\alpha_{\ell} \leftarrow \varphi(R_{\ell})$ // $G_4$ - $G_5$                                                                                                                 | 54 $\text{BAD}_2 \leftarrow \text{true}$ ; <b>abort</b> // $G_5$                                    |
| 23 <b>if</b> $\alpha_{\ell} \in \mathcal{S}_{\mathbb{A}}$ : $\text{BAD}_{\text{COLL}} := \text{true}$ ; <b>abort</b> // $G_4$ - $G_5$                                            | 55 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                                                        |
| 24 $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha_{\ell}\}$ // $G_4$ - $G_5$                                                                                | 56 <b>return</b> $\alpha$                                                                           |
| 25 $\text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha_{\ell}, \beta_{\ell})\}$ // $G_4$ - $G_5$                                                                              |                                                                                                     |
| 26 $s_{\ell} := k_{\ell}^{-1} \cdot (H(m_{\ell}) + x \cdot r_{\ell})$ // $G_1$ - $G_3$                                                                                           |                                                                                                     |
| 27 $\mathbf{U} := \mathbf{U} \  R_{\ell} \  R_{\ell}^{-1}$                                                                                                                       |                                                                                                     |
| 28 $\mathcal{S}_{sig} := \mathcal{S}_{sig} \cup m_{\ell}$                                                                                                                        |                                                                                                     |
| 29 <b>return</b> $(r_{\ell}, s_{\ell})$                                                                                                                                          |                                                                                                     |

**Fig. 35.** Games  $G_1$ - $G_5$  in the proof of Theorem 7.

**Game  $\mathbf{G}_2$ :** Now we exclude the case that  $\mathcal{A}$  has forged a signature without issuing a corresponding query to the bijective random oracles. For  $\mathcal{A}$ 's output  $(m^*, (r^*, s^*))$ , we first use the public key  $X$  to recover the value of  $R^*$  by computing  $(g^{H(m^*)} \cdot X^{r^*})^{1/s^*}$ . Then, if  $R^*$  has been added to the list  $\mathbf{U}$  (i.e.,  $\alpha^* \in \mathcal{S}_{\mathbb{A}}$ ), there must exist a corresponding pair  $(\varphi(R^*), \beta^*)$  recorded as the value of the function  $\Pi$  in the oracle list  $\text{LIST}_{\Pi}$ . Otherwise, we define the event  $\text{BAD}_1$  occurs and abort the game.

When  $\text{BAD}_1$  happens,  $\beta^*$  is chosen uniformly random and hidden from  $\mathcal{A}$ 's view. Therefore, for a fixed  $r^*$ , the winning condition  $r^* = \Psi(\beta^*)$  holds with probability at most  $\text{MAX}_{\Psi}/p$ . Since the distribution remains the same when  $\text{BAD}_1$  does not happen, we have

$$\left| \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}_1] \leq \frac{\text{MAX}_{\Psi}}{p}.$$

**Game  $\mathbf{G}_3$ :** In this game, we define an event  $\text{BAD}_{\text{COLL}0.5}$  occurs in the signing oracle  $\text{SIGN}$  if  $\alpha_{\ell} := \varphi(R_{\ell})$  has already been added to the list  $\text{LIST}_{\Pi}$ . Since  $R_{\ell}$  is generated by choosing uniformly at random an exponent from  $\mathbb{Z}_p$ ,  $\text{BAD}_{\text{COLL}0.5}$  occurs with probability at most  $q/p$ . Since  $\mathcal{A}$ 's view only changes when  $\text{BAD}_{\text{COLL}0.5}$  happens, we have

$$\left| \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] \right| \leq \Pr[\text{BAD}_{\text{COLL}0.5}] \leq \frac{q}{p}.$$

**Game  $\mathbf{G}_4$ :** In this game, we modify the way of simulating the signing oracle  $\text{SIGN}$  such that the secret key  $x$  is no longer required. Instead of first random choosing  $R_{\ell}$  and then computing  $\beta_{\ell} \leftarrow \Pi(\varphi(R_{\ell}))$  and  $r_{\ell} \leftarrow \Psi(\beta_{\ell})$  subsequently, we invert this computation by first sampling a fresh  $\beta_{\ell}$  uniformly at random and getting the corresponding  $r_{\ell}$  as the first component of the signature. Then, we choose the second component of the signature  $s_{\ell}$  uniformly at random, and use the verification equation to compute  $R_{\ell}$  as  $(g^{H(m_{\ell})} \cdot X^{r_{\ell}})^{1/s_{\ell}}$ . Finally, we use this value to reprogram the bijective random oracles by setting  $\Pi(\varphi(R_{\ell})) := \beta_{\ell}$ .

Since  $s_{\ell}$  is chosen uniformly random and independent of all other variables, the distribution of  $R_{\ell}$  in  $\mathbf{G}_4$  should also be uniformly random over  $\mathbb{G}$ . On the other hand, we define an event  $\text{BAD}_{\text{COLL}}$  in  $\mathbf{G}_4$ , which aborts the game if the function value  $\Pi(\varphi(R_{\ell}))$  has already been generated (i.e.,  $\alpha_{\ell} \in \mathcal{S}_{\mathbb{A}}$ ). This is actually equivalent to the event  $\text{BAD}_{\text{COLL}0.5}$  in  $\mathbf{G}_3$ . Therefore, our modification does not change the distribution, and hence

$$\Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1].$$

**Game  $\mathbf{G}_5$ :** This step is the same as the change from  $\mathbf{G}_2$  to  $\mathbf{G}_3$  in the proof of Lemma 2. That is, for each query  $\beta$  to the  $\text{BRO}^{-1}$ , we reprogram its corresponding preimage of  $\Pi$  as  $\alpha \leftarrow \varphi(R)$ , where the group element  $R = g^v X^w$  is generated by choosing uniformly at random two representations  $v, w \xleftarrow{\$} \mathbb{Z}_p$ . This modification does not change the distribution of the game, and hence

$$\Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1].$$

Same as the analysis of Lemma 2, every group element  $R \in \mathbf{U}$  now has a corresponding algebraic representation  $(v, w) \in \mathbb{Z}_p$  such that  $R = g^v X^w$ . Note that although we have an additional signing oracle  $\text{SIGN}$  in this proof, all the  $R_\ell$  generated in this oracle naturally have the algebraic representation  $(H(m_\ell) \cdot s_\ell^{-1}, r_\ell \cdot s_\ell^{-1})$ .

Now we use the reduction shown in Figure 36 to reduce the winning probability of  $\mathcal{A}$  in  $\mathbf{G}_5$  to the advantage of  $\mathcal{B}$  against the DL assumption.

| Reduction $\mathcal{B}(X)$                                                                                                                                                                  | Oracle $\text{BRO}(\vec{\rho})$                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 01 $\mathbf{U} := (g, X)$                                                                                                                                                                   | 18 $R \leftarrow \prod_i U_i^{\vec{\rho}_i}$                                                                                                    |
| 02 $(m^*, (r^*, s^*)) \xleftarrow{\$} \mathcal{A}^{\text{BRO}, \text{BRO}^{-1}, \text{SIGN}}(X)$                                                                                            | 19 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                                                                                                    |
| 03 <b>if</b> $m^* \in \mathcal{S}_{\text{sig}}$ : <b>return</b> $\perp$                                                                                                                     | 20 $\alpha := \varphi(R)$                                                                                                                       |
| 04 $R^* \leftarrow (g^{H(m^*)} \cdot X^{r^*})^{1/s^*}, \alpha^* := \varphi(R^*)$                                                                                                            | 21 <b>if</b> $\exists \beta$ s.t. $(\alpha, \beta) \in \text{LIST}_{\Pi}$ : <b>return</b> $\beta$                                               |
| 05 <b>if</b> $\alpha^* \notin \mathcal{S}_{\mathbb{A}}$                                                                                                                                     | 22 $\beta \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}, \mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta\}$   |
| 06 $\beta^* \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}, \text{BAD}_1 \leftarrow \text{true}; \text{abort}$                                                               | 23 $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha\}, \text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha, \beta)\}$      |
| 07 <b>else</b> fetch $\beta^*$ s.t. $(\alpha^*, \beta^*) \in \text{LIST}_{\Pi}$                                                                                                             | 24 <b>let</b> $(v, w)$ be $R = g^v X^w$                                                                                                         |
| 08 <b>let</b> $(v^*, w^*)$ be $R^* = g^{v^*} X^{w^*}$                                                                                                                                       | 25 <b>if</b> $w = -\Psi(\beta)$ : $\text{BAD}_2 \leftarrow \text{true}; \text{abort}$                                                           |
| 09 <b>return</b> $(H(m^*) - v^* s^*) / (w^* s^* - r^*)$                                                                                                                                     | 26 <b>return</b> $\beta$                                                                                                                        |
| Oracle $\text{SIGN}(m_\ell)$                                                                                                                                                                | Oracle $\text{BRO}^{-1}(\beta)$                                                                                                                 |
| 10 $s_\ell \xleftarrow{\$} \mathbb{Z}_p$                                                                                                                                                    | 27 <b>if</b> $\exists \alpha$ s.t. $(\alpha, \beta) \in \text{LIST}_{\Pi}$ : <b>return</b> $\alpha$                                             |
| 11 $\beta_\ell \xleftarrow{\$} \mathbb{B} \setminus \mathcal{S}_{\mathbb{B}}, \mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta_\ell\}, r_\ell \leftarrow \Psi(\beta_\ell)$ | 28 $\alpha \xleftarrow{\$} \mathbb{A} \setminus \mathcal{S}_{\mathbb{A}}, \mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha\}$ |
| 12 $R_\ell := (g^{H(m_\ell)} \cdot X^{r_\ell})^{1/s_\ell}, \alpha_\ell \leftarrow \varphi(R_\ell)$                                                                                          | 29 $\mathcal{S}_{\mathbb{B}} := \mathcal{S}_{\mathbb{B}} \cup \{\beta\}, \text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha, \beta)\}$       |
| 13 <b>if</b> $\alpha_\ell \in \mathcal{S}_{\mathbb{A}}$ : $\text{BAD}_{\text{COLL}} \leftarrow \text{true}; \text{abort}$                                                                   | 30 <b>if</b> $\alpha \in \varphi(\mathbb{G})$                                                                                                   |
| 14 $\mathcal{S}_{\mathbb{A}} := \mathcal{S}_{\mathbb{A}} \cup \{\alpha_\ell\}, \text{LIST}_{\Pi} := \text{LIST}_{\Pi} \cup \{(\alpha_\ell, \beta_\ell)\}$                                   | 31 $v, w \xleftarrow{\$} \mathbb{Z}_p, R := g^v X^w, \alpha \leftarrow \varphi(R)$                                                              |
| 15 $\mathbf{U} := \mathbf{U} \  R_\ell \  R_\ell^{-1}$                                                                                                                                      | 32 <b>if</b> $w = -\Psi(\beta)$ : $\text{BAD}_2 \leftarrow \text{true}; \text{abort}$                                                           |
| 16 $\mathcal{S}_{\text{sig}} := \mathcal{S}_{\text{sig}} \cup \{m_\ell\}$                                                                                                                   | 33 $\mathbf{U} := \mathbf{U} \  R \  R^{-1}$                                                                                                    |
| 17 <b>return</b> $(r_\ell, s_\ell)$                                                                                                                                                         | 34 <b>return</b> $\alpha$                                                                                                                       |

**Fig. 36.** Reduction to the DL assumption.

Since the game aborts when  $\alpha^* \notin \mathcal{S}_{\mathbb{A}}$ , the corresponding  $R^*$  must have been added to the list  $\mathbf{U}$  during oracle queries. This means that the simulator  $\mathcal{B}$  knows the algebraic representation of  $R^*$ . If  $\mathcal{A}$  wins and  $(w^* s^* - r^*) \neq 0$ , then  $\mathcal{B}$  can compute the value of  $x = (H(m^*) - v^* s^*) / (w^* s^* - r^*)$  as the solution to the DL instance.

Note that  $\mathcal{A}$  wins implies that  $r^* = \Psi(\beta^*)$ . Furthermore, if  $w^* s^* - r^* = 0$ , then from the verification equation we have  $H(m^*) - v^* s^* = 0$ , which implies

$$s^* = \frac{\Psi(\beta^*)}{w^*} = \frac{H(m^*)}{v^*}.$$

We now bound the probability of  $w^* s^* = \Psi(\beta^*)$ . Since  $\beta^*$  must have been generated during oracle queries, we divide the analysis into three cases:

- $\beta^*$  is generated in  $\text{BRO}$ : In this oracle, each fresh  $\beta$  is chosen uniformly at random and independent of  $R$  and  $m$ , and thus  $\Psi(\beta^*) = \frac{w^* \cdot H(m^*)}{v^*}$  holds with probability at most  $\text{MAX}_{\Psi}/p$ .

- $\beta^*$  is generated in  $\text{BRO}^{-1}$ : In this oracle, we sample  $(v^*, w^*)$  uniformly at random as the algebraic representation for a fresh  $R$ , which is independent of  $\beta$  and  $m$ , and thus  $\frac{w^*}{v^*} = \frac{\Psi(\beta^*)}{H(m^*)}$  holds with probability at most  $1/p$ .
- $\beta^*$  is generated in  $\text{SIGN}$ : This implies that there exists  $l \in [Q_S]$  such that  $r^* = r_\ell$ . Therefore, the representation of the corresponding  $R_\ell$  should be  $(H(m_\ell) \cdot s_\ell^{-1}, r_\ell \cdot s_\ell^{-1})$ , which implies  $s^* = s^l$  and  $H(m_\ell) = H(m^*)$ . Since  $m_\ell \neq m^*$ , this case can be bounded by the collision probability of the random oracle  $H$ .

Since  $\mathcal{B}$  wins when  $\mathcal{A}$  wins with  $w^*s^* - r^* \neq 0$ , we have

$$\begin{aligned} \Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] &\leq \text{Adv}^{\text{DL}}(\mathcal{B}) + \Pr[w^*s^* - r^* = 0 | \mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] \\ &\leq \text{Adv}^{\text{DL}}(\mathcal{B}) + q \cdot \frac{1 + \text{MAX}_\Psi}{p} + \frac{Q_H^2}{p}. \end{aligned}$$

Combining the above inequalities, we have

$$\text{Adv}_{Q_S, Q_H}^{\text{EUF-CMA}}(\mathcal{A}) \leq \text{Adv}^{\text{DL}}(\mathcal{B}) + \frac{2q + (q+1) \cdot \text{MAX}_\Psi + Q_H^2}{p}.$$

## F Strategy for Improving the Classic Security Bound of PQXDH

This section discusses how to extend our result to improve the classic security proof of PQXDH [42, 23]. Similar to  $\text{X3DH}^+$  we will include the context into the KDF and name the new protocol as  $\text{PQXDH}^+$ . The full code of PQXDH and  $\text{PQXDH}^+$  is presented in Figure 37. Note that here we do not consider the key-type domain separation of ephemeral and semi-static KEM public keys. In practice (and also in the specification of PQXDH [42]), the key-type domain separation should be included in the signatures of the KEM public keys.

To see how to apply our results to the PQXDH analysis [23], we first recall the security bounds of PQXDH in [23]. Note that we only consider the single-TEST security as in [23], where at most one session is queried to the TEST oracle. We translate the parameters of [23] into our notation: their  $n_p$ ,  $n_{ss}$ , and  $n_s$  correspond to our  $N_{\text{LT}}$ ,  $n_{\text{SS}}$ , and  $N_{\text{ses}}$ , respectively. In particular, their  $n_{ss}$  counts semi-static keys per user, whereas  $N_{\text{SS}} = N_{\text{LT}} \cdot n_{\text{SS}}$  counts semi-static keys across all users. We use this translation in both recalled bounds and the estimated bound below.

**POST-QUANTUM SECURITY.** We first discuss the estimated post-quantum security of  $\text{PQXDH}^+$ . The security bound of the post-quantum security of PQXDH in [23, Theorem 5.2] is recalled as follows:

$$\begin{aligned} \text{Adv}_{\text{PQXDH-Q}}^{\text{KI}}(\mathcal{A}) &\quad // \text{ Post-quantum security of PQXDH [23, Theorem 5.2]} \\ &\leq \frac{(N_{\text{LT}} + N_{\text{LT}} \cdot n_{\text{SS}} + N_{\text{ses}})^2}{p} + \gamma_{\text{coll}}(N_{\text{LT}} \cdot n_{\text{SS}} + N_{\text{ses}}) + N_{\text{ses}} \cdot \delta_{\text{corr}} \end{aligned} \quad (7)$$

| KGenLT(P)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | NewSession(P, role $\in \{I, R\}$ , type $\in \{\text{full}, \text{reduced}\}$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01 $isk_P := u \xleftarrow{\$} \mathbb{Z}_p$ , $ipk_P := U := g^u$<br>02 <b>return</b> ( $isk_P, ipk_P$ )<br>KGenSS( $isk_P = u$ )<br>03 $s \xleftarrow{\$} \mathbb{Z}_p$ , $S := g^s$<br>04 $\sigma \leftarrow \text{Sign}(u, S)$<br>05 $(sk_{ss}^{KEM}, pk_{ss}^{KEM}) \xleftarrow{\$} \text{KEM.KGen}()$<br>06 $\sigma_{ss}^{KEM} \leftarrow \text{Sign}(u, pk_{ss}^{KEM})$<br>07 $sssk := (s, sk_{ss}^{KEM})$<br>08 $sspk := ((S, \sigma), (pk_{ss}^{KEM}, \sigma_{ss}^{KEM}))$<br>09 <b>return</b> ( $sssk, sspk$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | // Initialize a session $\pi$<br>10 $r \xleftarrow{\$} \mathbb{Z}_p^*$ , $\rho_1 \xleftarrow{\$} \mathcal{R}_1, \rho_2 \xleftarrow{\$} \mathcal{R}_2, \rho_3 \xleftarrow{\$} \mathcal{R}_3$<br>11 $\pi.(\text{oid}, \text{role}) := (P, \text{role})$<br>12 <b>if</b> role = I<br>13 $\pi.\text{rand} := (r, \rho_3)$<br>14 <b>if</b> role = R<br>15 $\pi.\text{rand} := (r, \rho_1, \rho_2)$<br>16 <b>if</b> type = reduced: $\pi.\text{rand} := \perp$<br>17 $\pi.\text{type} := \text{type}$<br>18 <b>return</b> $\pi$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| InitR( $\pi, isk_B, ipk_B, sssk, sspk$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | InitI( $\pi, isk_A, ipk_A, m_{\text{bundle}}$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | DerR( $\pi, isk_B, ipk_B, sssk, sspk, m$ )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 19 $(b, B) := (isk_B, ipk_B)$<br>20 $(s, sk_{ss}^{KEM}) := sssk$<br>21 $((S, \sigma), (pk_{ss}^{KEM}, \sigma_{ss}^{KEM})) := sspk$<br>// Full mode<br>22 <b>if</b> $\pi.\text{type} = \text{full}$<br>23 $(y, \rho_1, \rho_2) := \pi.\text{rand}$<br>24 $Y := g^y$<br>25 $(sk_e^{KEM}, pk_e^{KEM}) := \text{KEM.KG}(\rho_1)$<br>26 $\sigma_e^{KEM} \leftarrow \text{Sign}(b, pk_e^{KEM}, \rho_2)$<br>27 $\tilde{Y} := (Y, pk_e^{KEM})$<br>// Reduced mode<br>28 <b>else</b><br>29 $\tilde{Y} := \perp, \sigma_e^{KEM} := \sigma_{ss}^{KEM}$<br>30 $B := \pi.\text{oid}$<br>31 $m_{\text{bundle}} := (B, (S, \sigma), (pk_{ss}^{KEM}, \sigma_{ss}^{KEM}), \tilde{Y}, \sigma_e^{KEM})$<br>32 Update $\pi$ :<br>33 $\pi.\text{pid} := \star$<br>// Unknown peer identity<br>34 $\pi.\text{state} := (y, Y, sk_e^{KEM}, pk_e^{KEM})$<br>35 $\pi.\text{cid} := (B, (S, \sigma), (pk_{ss}^{KEM}, \sigma_{ss}^{KEM}), \tilde{Y}, \sigma_e^{KEM})$<br>36 <b>return</b> ( $\pi, m_{\text{bundle}}$ ) | 37 $(a, A) := (isk_A, ipk_A)$<br>38 $(B, (S, \sigma), (pk_{ss}^{KEM}, \sigma_{ss}^{KEM}), \tilde{Y}, \sigma_e^{KEM}) := m_{\text{bundle}}$<br>39 $B := ipk[B]$<br>40 <b>if</b> $\text{Verify}(B, \sigma, S) \neq \text{true}$<br>41 <b>or</b> $\text{Verify}(B, \sigma_{ss}^{KEM}, pk_{ss}^{KEM}) \neq \text{true}$<br>42 <b>return</b> $\perp$<br>43 $(x, \rho_3) := \pi.\text{rand}$<br>44 $X := g^x$ , $\text{DH} := (S^a, B^x, S^x)$<br>45 <b>if</b> $\tilde{Y} \neq \perp$ // Full mode<br>46 $\pi.\text{type} := \text{full}$<br>47 $(Y, pk_e^{KEM}) := \tilde{Y}$<br>48 <b>if</b> $\text{Verify}(B, \sigma_e^{KEM}, pk_e^{KEM}) \neq \text{true}$<br>49 <b>return</b> $\perp$<br>50 $(ct^{KEM}, ss^{KEM}) := \text{Encap}(pk_e^{KEM}, \rho_3)$<br>51 <b>else</b> // Reduced mode<br>52 $(ct^{KEM}, ss^{KEM}) := \text{Encap}(pk_{ss}^{KEM}, \rho_3)$<br>53 // Session context<br>54 $\text{ctxt}_r := (A, X, B, S, \sigma, pk_{ss}^{KEM}, \sigma_{ss}^{KEM}, ct^{KEM})$<br>55 $\text{ctxt}_f := (A, X, B, S, Y, \sigma, pk_{ss}^{KEM}, \sigma_{ss}^{KEM}, pk_e^{KEM}, \sigma_e^{KEM}, ct^{KEM})$<br>56 // Key derivation<br>57 <b>if</b> $\pi.\text{type} = \text{full}$<br>58 $\text{DH} := \text{DH} \parallel Y^x$<br>59 $K := \text{KDF}_r(\text{DH}, \text{ctxt}_f, ss^{KEM})$<br>60 <b>else</b><br>61 $K := \text{KDF}_r(\text{DH}, \text{ctxt}_r, ss^{KEM})$<br>62 $\pi.(\text{pid}, \text{SK}) := (B, K)$<br>63 $\pi.\text{cid} := (B, (S, \sigma), (pk_{ss}^{KEM}, \sigma_{ss}^{KEM}), \tilde{Y}, \sigma_e^{KEM})$<br>64 $\pi.\text{sid} := (A, X, B, S, Y, \sigma, pk_{ss}^{KEM}, \sigma_{ss}^{KEM}, pk_e^{KEM}, \sigma_e^{KEM}, ct^{KEM})$<br>65 <b>return</b> ( $\pi, m := (A, X, ct^{KEM})$ ) | 66 $B := \pi.\text{oid}$<br>67 $(b, B) := (isk_B, ipk_B)$<br>68 $(s, sk_{ss}^{KEM}) := sssk$<br>69 $((S, \sigma), (pk_{ss}^{KEM}, \sigma_{ss}^{KEM})) := sspk$<br>70 $(y, Y, sk_e^{KEM}, pk_e^{KEM}) := \pi.\text{state}$<br>71 $(A, X, ct^{KEM}) := m$<br>72 $\text{ctxt}_r := (A, X, B, S, \sigma, pk_{ss}^{KEM}, \sigma_{ss}^{KEM}, ct^{KEM})$<br>73 $\text{ctxt}_f := (A, X, B, S, Y, \sigma, pk_{ss}^{KEM}, \sigma_{ss}^{KEM}, pk_e^{KEM}, \sigma_e^{KEM}, ct^{KEM})$<br>74 $\text{DH} := (S^s, X^b, X^s)$<br>75 <b>if</b> $\pi.\text{type} = \text{full}$<br>76 $\text{DH} := \text{DH} \parallel X^y$<br>77 $ss^{KEM} \leftarrow \text{Decap}(sk_{ss}^{KEM}, ct^{KEM})$<br>78 $K := \text{KDF}_r(\text{DH}, \text{ctxt}_f, ss^{KEM})$<br>79 <b>else</b><br>80 $ss^{KEM} \leftarrow \text{Decap}(sk_{ss}^{KEM}, ct^{KEM})$<br>81 $K := \text{KDF}_r(\text{DH}, \text{ctxt}_r, ss^{KEM})$<br>82 $\pi.(\text{pid}, \text{SK}) := (A, K)$<br>83 $\pi.\text{sid} := (A, X, B, S, Y, \sigma, pk_{ss}^{KEM}, \sigma_{ss}^{KEM}, pk_e^{KEM}, \sigma_e^{KEM}, ct^{KEM})$<br>84 <b>return</b> $\pi$ |

**Fig. 37.** Protocol PQXDH<sup>+</sup>. The dark gray part indicates the additional post-quantum KEM comparing to X3DH<sup>+</sup>. For convenience, here we do not consider the key-type domain separation (e.g., include labels “PQXDH-PQOPK” and “PQXDH-PQSPK” in the signatures of ephemeral and semi-static post-quantum public keys).

$$\begin{aligned}
& + \epsilon_{\text{LEAK}+r} && // \text{ Binding properties of KEM} \\
& + n_{\text{SS}} \cdot N_{\text{ses}} \cdot (\epsilon_{\text{CCA}} + \epsilon_{\text{PRF}}) && // \text{ clean}_{\text{Q-ESS}}^{\text{pqxdh}} \\
& + N_{\text{ses}}^2 \cdot (\epsilon_{\text{CCA}} + \epsilon_{\text{PRF}}). && // \text{ clean}_{\text{Q-EE}}^{\text{pqxdh}}
\end{aligned}$$

The analysis of PQXDH considers post-quantum counterparts of the predicates Eph-Eph and Eph-SS, denoted by  $\text{clean}_{\text{Q-EE}}^{\text{pqxdh}}$  and  $\text{clean}_{\text{Q-ESS}}^{\text{pqxdh}}$ , respectively. We leave their formal definitions to [23, Figure 9]. Both  $\text{clean}_{\text{Q-ESS}}^{\text{pqxdh}}$  and  $\text{clean}_{\text{Q-EE}}^{\text{pqxdh}}$  do not consider any classic security of PQXDH. This is because for post-quantum adversaries, all DH secrets provide no security. Hence, our result on X3DH<sup>+</sup> cannot be applied for the post-quantum security of PQXDH.

Therefore, we can conclude that the post-quantum security of PQXDH<sup>+</sup> is the same as the post-quantum security of PQXDH in [23, Theorem 5.2]. Namely, the post-quantum security bound of PQXDH<sup>+</sup> should be the same as  $\text{Adv}_{\text{PQXDH-Q}}^{\text{KI}}(\mathcal{A})$  (as recalled in Equation (7)).

**CLASSIC SECURITY OF PQXDH<sup>+</sup>.** We now move on to estimate the classic security bound of PQXDH<sup>+</sup> based on the security bound of PQXDH. The security bound of the classic security of PQXDH in [23, Theorem 5.1] is recalled as follows.

$$\begin{aligned}
& \text{Adv}_{\text{PQXDH}}^{\text{KI}}(\mathcal{A}) && // \text{ Classic security of PQXDH [23, Theorem 5.1]} && (8) \\
& \leq \frac{(N_{\text{LT}} + N_{\text{LT}} \cdot n_{\text{SS}} + N_{\text{ses}})^2}{p} + \gamma_{\text{coll}}(N_{\text{LT}} \cdot n_{\text{SS}} + N_{\text{ses}}) + N_{\text{ses}} \cdot \delta_{\text{corr}} \\
& + \epsilon_{\text{LEAK}+r} && // \text{ Binding properties of KEM} \\
& + N_{\text{LT}} \cdot (\epsilon_{\text{SIG}} + N_{\text{LT}} \cdot n_{\text{SS}} \cdot \epsilon_{\text{GDH}}) && // \text{ clean}_{\text{LTSS}}^{\text{pqxdh}} \\
& + N_{\text{ses}} \cdot N_{\text{LT}} \cdot \epsilon_{\text{GDH}} && // \text{ clean}_{\text{ELT}}^{\text{pqxdh}} \\
& + N_{\text{LT}} \cdot (\epsilon_{\text{SIG}} + n_{\text{SS}} \cdot N_{\text{ses}} \cdot \epsilon_{\text{GDH}}) && // \text{ clean}_{\text{ESS}}^{\text{pqxdh}} \wedge \text{type} = \text{full} \\
& + N_{\text{LT}} \cdot (\epsilon_{\text{SIG}} + n_{\text{SS}} \cdot N_{\text{ses}} \cdot \min(\epsilon_{\text{GDH}}, \epsilon_{\text{CCA}})) && // \text{ clean}_{\text{ESS}}^{\text{pqxdh}} \wedge \text{type} = \text{reduced} \\
& + N_{\text{ses}}^2 \cdot \min(\epsilon_{\text{GDH}}, \epsilon_{\text{CCA}}) && // \text{ clean}_{\text{EE}}^{\text{pqxdh}} \wedge \text{clean}_{\text{peerE}}^{\text{pqxdh}} \\
& + N_{\text{LT}} \cdot (\epsilon_{\text{SIG}} + N_{\text{ses}}^2 \cdot Q_{\text{kdf}} \cdot \epsilon_{\text{CCA}}). && // \text{ clean}_{\text{EE}}^{\text{pqxdh}} \wedge \text{clean}_{\text{sigE}}^{\text{pqxdh}}
\end{aligned}$$

Below, we discuss the strategy of how to extend our proof of X3DH<sup>+</sup> to improve the classic security proof of PQXDH<sup>+</sup>:

1. Similar to the tight security proof of Theorem 1, we first use the game hop  $\mathbf{G}_1$ - $\mathbf{G}_6$  (cf. Figure 16) to exclude the cases that the adversary computes any fresh DH secret or forges the signature on the semi-static public key (including the post-quantum KEM's semi-static and ephemeral public keys). After this step, we introduce the bound Equation (2) into the final security bound of PQXDH<sup>+</sup>. Note that our technique removes the guessing security loss on signature in Equation (8), due to our multi-user with corruptions security notion.
2. After the above step, one can then perform the same analysis steps in the proof of [23, Theorem 5.1] to analyze the remaining cases, except that all

Diffie-Hellman reductions are ignored (because they are already addressed in the bound Equation (2)).

Therefore, the classical security bound of PQXDH<sup>+</sup> appears to be the following

$$\begin{aligned}
& \text{Adv}_{\text{PQXDH}^+}^{\text{KI}}(\mathcal{A}) \quad // \text{ Estimated classic security bound of PQXDH}^+ \\
& \leq \epsilon_{\text{sign-ddh}} + \epsilon_{\text{CorrGapCDH}} + N_{\text{LT}} \cdot \delta_{\text{sim}} + \frac{(N_{\text{LT}} + N_{\text{SS}} + N_{\text{ses}})^2}{2p} \\
& \quad + \gamma_{\text{coll}} \cdot (N_{\text{LT}} \cdot n_{\text{SS}} + N_{\text{ses}}) + N_{\text{ses}} \cdot \delta_{\text{corr}} + \epsilon_{\text{LEAK}^+} \\
& \quad + N_{\text{LT}} \cdot n_{\text{SS}} \cdot N_{\text{ses}} \cdot \epsilon_{\text{CCA}} + N_{\text{ses}}^2 \cdot \epsilon_{\text{CCA}} + N_{\text{LT}} \cdot N_{\text{ses}}^2 \cdot Q_{\text{kdf}} \cdot \epsilon_{\text{CCA}},
\end{aligned}$$

where  $\epsilon_{\text{sign-ddh}}$  and  $\epsilon_{\text{CorrGapCDH}}$  are the same as the first two terms in Equation (2).

However, the aforementioned bounds are our estimates. We leave the detailed implementation of our strategy as a future work.

## G Use of AI Tools

The authors used ChatGPT (GPT-5.6 Sol) and OpenAI Codex (GPT-6 Astra) to assist with revising this manuscript. ChatGPT reviewed manuscript material and suggested corrections to language, notation, references, pseudocode, and consistency between theorem statements and proofs. The authors revised the manuscript themselves based on ChatGPT's suggestions and also passed some of the suggestions to Codex, which assisted with implementing requested revisions in the LaTeX codes. The authors reviewed and double-checked the changes made by Codex.

The assistance included spelling and grammar corrections, consolidation of duplicate bibliography entries, and corrections to cross-references and figure captions. The tools also helped check and correct inconsistencies in mathematical notation, algorithm inputs and variable usage, and references to security games.

Responsibility for the final manuscript remains with the authors.