# Foundations of Dynamic Group Signatures: The Case of Malicious Openers and Issuers

Stephan Krenn[1], Kai Samelin[2], and Daniel Slamanig[3]

[1] AIT Austrian Institute of Technology, Vienna, Austria
stephan.krenn@ait.ac.at
[2] Independent, Hamburg, Germany
kaispapers@gmail.com
[3] Research Institute CODE, Universität der Bundeswehr München, München, Germany
daniel.slamanig@unibw.de

**Abstract.** Group signatures enable users to sign on behalf of a group while preserving anonymity, with accountability provided by a designated opener. The first rigorous model for dynamic groups (Bellare, Shi, Zhang, CT–RSA '05) captured anonymity, non-frameability, and traceability, later extended with trace-soundness (Sakai et al., PKC '12) and non-claimability (introduced as "opening-soundness" by Bootle et al., ACNS '16 & JoC '20).

In practice, issuer and opener are often distinct entities, often implemented by different organizations and/or hardware modules. We therefore formalize and prove the consequences of a model that enforces their complete separation, allows key reuse across groups, treats issuer and opener as stateless, and makes both joining and opening non-interactive.

This separation makes it necessary to reformulate traceability against a corrupt issuer and to introduce three additional unforgeability notions—key-unforgeability, certificate-unforgeability, and opening-unforgeability—for the case of a corrupt opener. Following this line of reasoning, we also develop strengthened formulations of trace-soundness and non-claimability.

We prove that in this model the eight resulting properties are fully distinct: even the conjunction of any seven does not imply the eighth. This yields the first comprehensive map of group signature security in a stateless, reusable-key, and non-interactive framework, and formally demonstrates the impact of complete issuer–opener separation.

**Keywords:** Group Signatures ◇ Traceability ◇ Security Models ◇ Framework

## 1 Introduction

Group signatures, introduced by Chaum and van Heyst [13], allow users to anonymously sign messages on behalf of a group. A designated opener can later revoke this anonymity and identify the signer in case of abuse. This balance between privacy and accountability makes group signatures an important primitive for applications such as electronic IDs, e-voting, and privacy-preserving access control.

**From BSZ to Bootle et al.** The first rigorous model for dynamic groups was given by Bellare, Shi, and Zhang (called BSZ) [5], capturing *anonymity*, *non-frameability*,

| Reference | Monolithic | Issuer | Opener | Comment |
|---|---|---|---|---|
| Bellare et al. [5] | ✗ | honest | leaked key | Initial BSZ definition |
| Bichsel et al. [7] | ✓ | honest | honest | Monolithic authority |
| Sakai et al. [23] | ✗ | honest | leaked key | Added trace-soundness |
| Bootle et al. [10] | ✗ | honest | $\mathcal{A}$ | Non-claimability; honest issuer |
| Bootle et al. [11] | ✗ | honest | leaked key | Reverted to BSZ-style |
| Boneh et al. [9] | ✓ | honest | honest | Monolithic |
| Backes et al. [2] | ✗ | honest | $\mathcal{A}$ | Honest issuer only |
| Groth [19] | ✗ | honest | leaked key | BSZ-style |
| Benhamouda et al. [6] | ✗ | honest | leaked key | BSZ-style |
| Delerablée et al. [14] | ✗ | honest | leaked key | BSZ-style |
| Alamelou et al. [1] | ✗ | honest | honest | Both honest |
| Krenn et al. [21] | ✗ | honest | honest | Both honest |
| Bellare et al. [4] | ✓ | honest | leaked key | Monolithic; no joins |
| Derler&Slamanig [16] | ✗ | honest | leaked key | Initial BSZ definition |

Table 1: Overview of traceability definitions. “Monolithic” means issuer and opener are identical. “$\mathcal{A}$” means adversary-generated keys. “Leaked key” means the opener key is given to the adversary. All schemes assume an honest issuer.

and *traceability*. Sakai et al. [23] added *trace-soundness*, while Bootle et al. [10,11] introduced *non-claimability* (initially termed “opening-soundness”). Yet, across these definitions the roles of issuer (who decides which users are part of the group) and opener were never fully disentangled: BSZ-style traceability presumes both keys are honestly generated while leaking the opener’s secret to the adversary.[4] Other works even collapse issuer and opener into a monolithic authority or assume trusted key generation. This results in a fragmented landscape of definitions; see Table 1.

**Practical Separation.** In practice, issuer and opener are rarely the same entity: the former may be a certificate authority, while the latter could be realized by a hardware security module or even a public authority. These realities motivate a model that enforces *complete issuer–opener separation*, and supports key *reuse* across groups. Note, a group is uniquely determined by the tuple of the public keys from the issuer and opener. We treat both roles as *stateless*, rendering joining and opening non-interactive.

**Our Main Contributions.** Building on the five established properties of group signatures—*anonymity* (signer cannot be identified), *non-frameability* (a user cannot be blamed for a signature it did not create), *traceability* (every verifying signature can be opened), *trace-soundness* (at most one opening exists), and *non-claimability* (no re-

[4] However, as Bellare et al. explicitly state, “Traceability asks that the adversary be unable to produce a signature such that either *the honest opener* declares itself unable to identify the origin of the signature [...] or, the honest opener believes it has identified the origin but is unable to produce a correct proof of its claim [...] but the proof [...] is rejected by the judge.”. [5]

| **Property** | **Issuer** | **Opener** | **Honest Members** | **Goal** |
|---|---|---|---|---|
| Anonymity | ✗ | ✓ | $\geq 2!$ | Hides the signer's identity |
| Non-frameability | ✗ | ✗ | $\geq 1$ | No false attribution to honest members |
| Traceability | ✗ | ✓! | ✗ | Every valid signature can be opened |
| Trace-soundness | ✗ | ✗ | ✗ | No signature admits two different openings |
| Non-claimability | ✗ | ✗ | $\geq 1!$ | Signatures cannot be reattributed |
| Cert-unforgeability | ✓ | ✗ | all! | No signatures without certificate |
| Key-unforgeability | ✓ | ✗ | all | No signatures without secret key |
| Opening-unforgeability | ✓ | ✗ | ✗ | No valid opening for a non-member |

Table 2: Honesty assumptions for our seven properties. ✓ means the role is honest, ✗ means it may be malicious, while a trailing ! indicates that the honestly generated secrets are given to $\mathcal{A}$. For cert-unforgeability, the membership certificates are given to the adversary.

attribution of signatures)—we develop a refined model that enforces complete issuer–opener separation and analyze its consequences:

- We strengthen *traceability* to hold even against a malicious issuer.
- We introduce three new unforgeability notions to address *malicious* openers: *certificate-unforgeability* (preventing signatures without a valid certificate), *key-unforgeability* (preventing signatures without the corresponding secret key), and *opening-unforgeability* (preventing openings to non-members).
- Following the same reasoning, we also develop strengthened formulations of *trace-soundness* (an opening is only valid for one group and user) and *non-claimability* (we also give the adversary the user's secret key).

Together with anonymity and non-frameability, this yields a framework of eight properties. Table 2 summarizes the honesty assumptions underlying each notion.

**Independence.** We prove that the eight resulting properties are fully independent: even the conjunction of any seven does not imply the eighth. This shows that the eight properties form a core framework once issuer and opener are fully separated.

Our ideas also immediately apply to fully dynamic and revocable group signature schemes, since these can be viewed as extensions of the our ideas. A detailed treatment is left for future work.

**A Concrete Example.** The need for such refinements is not merely theoretical: concrete schemes have been shown to fail once the opener misbehaves. Sakai et al. [23] already show that a malicious opener controlling two members (only one of them needs to be joined) in the construction by Furukawa and Imai [18] can produce openings which point to a member outside the group - which is covered by our opening unforgeability definition, even though they motivate this behavior for their trace-soundness definition.

In more detail, in the scheme of Furukawa and Imai [18], let $(R, V) = (g^{x_i + r}, S^r)$ be the ElGamal ciphertext inside a signature by user $i$. The honest opener produces a valid opening $O_i$ pointing to $Q_i = g^{x_i}$. If the opener is malicious and controls members $i, j$, then reusing $r$ yields a forged opening $O_j$ for $Q_j = g^{x_j}$ with witness $w = sr/(x_i + r - x_j)$. Thus, from a valid $(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m)$ in the list $\mathcal{Q}$ of honest-user signatures, the adversary outputs $(\mathsf{opk}, Q_j, m, \sigma_m, O_j)$ with $(\mathsf{opk}, Q_j)$ never joined, yet the Judge algorithm will accept. This matches our definition of an *opening unforgeability* break.

**Compatibility with BSZ-style Schemes.** We further provide a BSZ-style construction (without distributed setup) that achieves all eight properties under standard assumptions. In doing so, we highlight two additional ingredients that are required in the *sign–encrypt–prove* paradigm in our setting: (i) *decryption soundness*, ensuring honest ciphertexts decrypt consistently under any valid decryption key, and (ii) *statement-binding*, a NIZK meta-property preventing ambiguous proofs.

## 2 Preliminaries

**Notation.** Let $\lambda \in \mathbb{N}$ denote the security parameter. Unless explicitly stated otherwise, all algorithms implicitly receive $1^\lambda$ as auxiliary input. We write $a \leftarrow_\$ A(x)$ to denote that $a$ is the output of a probabilistic algorithm $A$ on input $x$. If $A$ is deterministic, we write $a \leftarrow A(x)$. An algorithm is *efficient* if it runs in probabilistic polynomial time (PPT) in $\lambda$ and in the length of its explicit inputs; unless noted otherwise, all algorithms in this work are assumed to be PPT. If an algorithm $A$ uses external coins, we write $a \leftarrow_\$ A(x; \xi)$, where $\xi$ denotes its randomness. If output and coins are relevant, we write $(a; \xi) \leftarrow_\$ A(x)$. Algorithms may output the distinguished error symbol $\perp \notin \{0,1\}^*$ to indicate failure or an exceptional condition; returning $\perp$ aborts the corresponding algorithm or oracle. We use $(a, \perp) \leftarrow_\$ A(x)$ to emphasize that a second component is unused.

If $S$ is a finite set, then $a \leftarrow_\$ S$ means that $a$ is sampled uniformly from $S$. We denote by $\mathcal{M}$ the message space of a cryptographic scheme, assumed to be efficiently derivable from the scheme's public parameters or public key. For lists/tuples, we assume an injective, efficiently invertible encoding into $\{0,1\}^*$.

A function $\nu : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if for every polynomial $p(\cdot)$ there exists $n_0 \in \mathbb{N}$ such that for all $n > n_0$, $\nu(n) \leq 1/p(n)$.

**Building Blocks.** We require one-way functions and unforgeable digital signatures. Additionally, we require mcIND-CPA [3], decryption-sound, key-verifiable public key encryption and non-interactive simulation-sound extractable zero-knowledge proof systems with statement-binding. As decryption-soundness, key-verifiability and statement-binding are non-standard requirements, we present them here. All other formal definitions are given in Appendix A.

**Decryption-Sound Key-Verifiable Public Key Encryption Schemes.** We require a public-key encryption (PKE) scheme (denoted $\Theta$) with two non-standard, but natural, properties: *key verifiability* and *decryption soundness*.

*Key verifiability* ensures that a given decryption key $\mathsf{sk}_\Theta$ can be verified as consistent with a public encryption key $\mathsf{pk}_\Theta$. This is necessary because $\mathsf{sk}_\Theta$ appears as part of a NIZK witness and must therefore be externally validated.

*Decryption soundness* ensures that for any ciphertext $c$, decryption under a (key-verified) key pair $(\mathsf{pk}_\Theta, \mathsf{sk}_\Theta)$ yields the originally encrypted plaintext. That is, it should be infeasible to produce $(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta, m^*, c)$ with $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1$ and $c \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, m^*)$ but $\mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c) \neq m^*$. This is crucial since opening relies on correct decryption.

Both properties are essential. In particular, *decryption soundness closes a gap not covered by key verifiability alone*: even if $\mathsf{sk}_\Theta$ matches $\mathsf{pk}_\Theta$, a malicious adversary could try to craft ciphertexts that decrypt inconsistently. Our proofs rely on *both* properties.

We formalize both notions below and show that ElGamal over *prime-order* groups suffices.

The standard definitions (framework, correctness, and mcIND-CPA) are presented in Appendix A.

**Key-Verifiability.** A PKE $\Theta$ is key-verifiable if no efficient adversary can produce two decryption keys that both pass $\mathsf{KVfy}_\Theta$ for the same public key $\mathsf{pk}_\Theta$ yet yield *different* decryptions of some ciphertext.

```
$\mathbf{Exp}^{\mathsf{KeyVerifiability}}_{\mathcal{A},\Theta}(\lambda)$ :
  $\mathsf{pp}_\Theta \leftarrow_\$ \mathsf{PP}_\Theta(1^\lambda)$
  $(\mathsf{sk}^*_{\Theta 0}, \mathsf{sk}^*_{\Theta 1}, \mathsf{pk}^*_\Theta, c^*) \leftarrow_\$ \mathcal{A}(\mathsf{pp}_\Theta)$
  return $0$ if $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_{\Theta 0}, \mathsf{pk}^*_\Theta) \neq 1 \ \vee \ \mathsf{KVfy}_\Theta(\mathsf{sk}^*_{\Theta 1}, \mathsf{pk}^*_\Theta) \neq 1$
  return $1$ if $\mathsf{Dec}_\Theta(\mathsf{sk}^*_{\Theta 0}, c^*) \neq \mathsf{Dec}_\Theta(\mathsf{sk}^*_{\Theta 1}, c^*)$
  return $0$
```

Fig. 1: Key Verifiability of a PKE $\Theta$

**Definition 1 (Key Verifiability).** *A PKE $\Theta$ offers key verifiability if for every PPT adversary $\mathcal{A}$ there exists a negligible $\nu$ such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{KeyVerifiability}}_{\mathcal{A},\Theta}(\lambda) = 1\right] \leq \nu(\lambda),$$

*where the corresponding experiment is given in Figure 1.*

**Decryption-Soundness.** A PKE $\Theta$ is *decryption-sound* if no adversary can produce a tuple $(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta, m^*)$ with $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1$ such that a fresh ciphertext $c \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, m^*)$ decrypts to a different value under $\mathsf{sk}^*_\Theta$. This mirrors the key-verifiability refinement from [15] in our setting.

**Definition 2 (Decryption Soundness).** *A PKE $\Theta$ is decryption-sound if for every PPT adversary $\mathcal{A}$ there exists a negligible $\nu$ such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{DecSound}}_{\mathcal{A},\Theta}(\lambda) = 1\right] \leq \nu(\lambda),$$

```
$\mathbf{Exp}^{\mathsf{DecSound}}_{\mathcal{A},\Theta}(\lambda)$ :
    $\mathsf{pp}_\Theta \leftarrow_\$ \mathsf{PP}_\Theta(1^\lambda)$
    $(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta, m^*, r^*) \leftarrow_\$ \mathcal{A}(\mathsf{pp}_\Theta)$
    return $0$ if $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) \neq 1$
    $c \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, m^*; r^*)$
    return $0$ if $c = \bot$
    return $1$ if $\mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c) \neq m^*$
    return $0$
```

Fig. 2: Decryption Soundness of a PKE $\Theta$

*where the corresponding experiment is given in Figure 2.*

We stress that decryption soundness is not equivalent to correctness. While correctness states that for every honestly generated key pair and ciphertext, decryption yields the original plaintext, the notion of decryption soundness also covers potentially rogue keys generated by an adversary, and only requires the ciphertext to be honestly generated. While this seems like a subtlety, one can easily build perfectly correct and key verifiable schemes that do not satisfying decryption soundness by simply letting key verification return 1 for keys that can never be output by an honest run of the key generation algorithm. Concrete counterexamples will be given later in this section.

Schemes that intentionally have multiple secret keys, potentially mapping to different outputs, appear in non-committing encryption, functional encryption and related primitives [8,12,20,22].

We provide the concrete instantiation for reference in Appendix A. For the following theorems let $\Theta$ be the standard ElGamal PKE in prime-order groups.

**Theorem 1.** *$\Theta$ is key-verifiable.*

*Intuition.* As $g$ is a generator of a prime-order $q$ group and we work in the exponent modulo $q$, there is exactly one secret key for each public key w.r.t. $g$.

The proof is given in Appendix B.1.

**Theorem 2.** *$\Theta$ is decryption-sound.*

*Intuition.* Proof follows directly from the algebraic structure of ElGamal.

The proof is given in Appendix B.2.

**Theorem 3 (Key-Verifiability $\not\Longrightarrow$ Decryption-Soundness).** *Key-Verifiability does not imply Decryption-Soundness.*

*Intuition.* The idea behind the proof is to add a tag $t$ to a public key and always return $t$, if $t \neq \bot$.

The proof is given in Appendix B.3.

**Theorem 4 (Decryption Soundness $\not\Rightarrow$ Key Verifiability).** *There exists a correct PKE $\Theta''$ that is decryption-sound but not key-verifiable.*

*Intuition.* The idea behind the proof is to add a flag $f$ to the secret key and the ciphertext. Depending on $f$, the decryption yields different results.

The proof is given in Appendix B.4.

**Non-Interactive Proof Systems.** We require that the statement $x$ captures all public parameters relevant to the relation (public keys, ciphertexts, message, ...). Thus $x$ fixes the intended meaning, and SSE holds with respect to the full statement. This can be enforced by appending the statement to the proof [17].

**Statement-Binding.** We formalize this requirement by introducing *Statement-Binding*: a single proof $\pi$ should not verify for two different statements. We require this notion to prove our strengthened trace-soundness notion; See Section 3.

```
$\mathbf{Exp}^{\mathsf{StatementBinding}}_{\mathcal{A},\Pi,\mathcal{E}}(\lambda)$ :
  $(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_\$ \mathcal{E}(1^\lambda)$
  $(x_0^*, x_1^*, \pi^*) \leftarrow_\$ \mathcal{A}(\mathsf{crs}_\Pi, \tau, \zeta)$
  return $0$ if $x_0^* = x_1^*$
  return $1$ if $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x_0^*, \pi^*) = 1 \ \wedge \ \mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x_1^*, \pi^*) = 1$
  return $0$
```

Fig. 3: Statement-Binding of a NIZK $\Pi$

**Definition 3 (Statement-Binding).** *A NIZK $\Pi$ satisfies Statement-Binding if for every PPT adversary $\mathcal{A}$ there exists an extractor $\mathcal{E}$ and a negligible function $\nu$ s.t.*

$$\Pr\left[\mathbf{Exp}^{\mathsf{StatementBinding}}_{\mathcal{A},\Pi,\mathcal{E}}(\lambda) = 1\right] \leq \nu(\lambda).$$

**Construction of Statement-Binding $\Pi$s.** We construct $\Pi'$ that is zero-knowledge, simulation-sound extractable, and statement-binding from any $\Pi$ with ZK and SSE.

Let $\Pi$ be ZK and SSE. Consider $\Pi'$ as follows.

**Security.** We prove ZK, SSE, and Statement-Binding for $\Pi'$.

**Theorem 5 (Zero-Knowledge of $\Pi'$).** *If $\Pi$ is zero-knowledge, then $\Pi'$ is zero-knowledge.*

$\mathsf{PG}'_\Pi$**.** As in $\Pi$: $\mathsf{crs}_\Pi \leftarrow_\$ \Pi.\mathsf{PG}_\Pi(1^\lambda)$.
$\mathsf{Prf}'_\Pi$**.** On $(\mathsf{crs}_\Pi, x, w)$ compute $\pi' \leftarrow_\$ \Pi.\mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, x, w)$ and output $\pi \leftarrow (\pi', x)$.
$\mathsf{Vfy}'_\Pi$**.** On $(\mathsf{crs}_\Pi, x, \pi)$ parse $\pi = (\pi', x')$. If $x \neq x'$ return 0; else return $\Pi.\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x, \pi')$.

Construction 1: Statement-Binding $\Pi'$

*Proof.* Let $\mathsf{SIM} = (\mathsf{SIM}_1, \mathsf{SIM}_2)$ be the simulator for $\Pi$ and define $\mathsf{SIM}'_1 = \mathsf{SIM}_1$, $\mathsf{SIM}'_2(\mathsf{crs}_\Pi, \tau, x) = (\mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, x), x)$. Since both real and ideal worlds include $x$ verbatim, the distinguishing advantage equals that for $\Pi$ and is negligible.

**Theorem 6 (Simulation-Sound Extractability of $\Pi'$).** *If $\Pi$ is simulation-sound extractable, then $\Pi'$ is simulation-sound extractable.*

*Proof.* Let $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ be the extractor for $\Pi$ and set $\mathcal{E}'_1 = \mathcal{E}_1, \mathcal{E}'_2(\mathsf{crs}_\Pi, \zeta, x, (\pi', x')) \leftarrow \perp$ if $x \neq x'$, else $\mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, x, \pi')$. If $x \neq x'$, verification under $\mathsf{Vfy}'_\Pi$ already fails; if $x = x'$, SSE of $\Pi$ gives extraction except with negligible probability.

**Theorem 7 (Statement-Binding of $\Pi'$).** *$\Pi'$ satisfies Statement-Binding.*

*Proof.* If $\pi = (\pi', x')$ verifies for both $x_0$ and $x_1$, then $\mathsf{Vfy}'_\Pi$ enforces $x_0 = x' = x_1$, contradicting $x_0 \neq x_1$.

## 3 Group Signature Schemes

We now present the formal definitions of a group signature scheme (denoted $\Delta$). Our formalization follows the framework by Bellare, Shi, and Zhang (BSZ) [5], augmented with refinements introduced in later works such as Sakai et al. [23] and Bootle et al. [11]. In particular, we make explicit distinctions between membership certificates and signatures, and account for properties that became necessary in subsequent analyses.

We then present a contrived scheme that is secure in the respective model, but nevertheless fails to provide the security guarantees one intuitively expects.

**Framework.** We now present the framework.

**Definition 4 (Group Signature Scheme).** *A group signature scheme $\Delta$ is a tuple of PPT algorithms*

$$(\mathsf{PG}_\Delta, \mathsf{KgM}_\Delta, \mathsf{KgI}_\Delta, \mathsf{KgO}_\Delta, \mathsf{Join}_\Delta, \mathsf{Sgn}_\Delta, \mathsf{Vfy}_\Delta, \mathsf{Opn}_\Delta, \mathsf{Jdg}_\Delta),$$

*with the following interfaces (where $\sigma_s$ denotes a membership certificate and $\sigma_m$ a group signature):*

$\mathsf{PG}_\Delta$**.** *On input $1^\lambda$ output public parameters $\mathsf{pp}_\Delta$. We assume $\mathsf{pp}_\Delta$ is implicit input to all other algorithms:*

$$\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda).$$

$\mathsf{KgM}_\Delta$. *On input* $\mathsf{pp}_\Delta$*, output a member secret/public key pair* $(\mathsf{usk}, \mathsf{upk})$*:*

$$(\mathsf{usk}, \mathsf{upk}) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta).$$

$\mathsf{KgI}_\Delta$. *On input* $\mathsf{pp}_\Delta$*, output issuer keys* $(\mathsf{isk}, \mathsf{ipk})$*:*

$$(\mathsf{isk}, \mathsf{ipk}) \leftarrow_\$ \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta).$$

$\mathsf{KgO}_\Delta$. *On input* $\mathsf{pp}_\Delta$*, output opener keys* $(\mathsf{osk}, \mathsf{opk})$*:*

$$(\mathsf{osk}, \mathsf{opk}) \leftarrow_\$ \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta).$$

$\mathsf{Join}_\Delta$. *On input the issuer secret key* $\mathsf{isk}$ *and public keys* $\mathsf{upk}, \mathsf{opk}$*, output a membership certificate* $\sigma_s$*:*

$$\sigma_s \leftarrow_\$ \mathsf{Join}_\Delta(\mathsf{isk}, \mathsf{upk}, \mathsf{opk}).$$

$\mathsf{Sgn}_\Delta$. *On input a member secret key* $\mathsf{usk}$*, a certificate* $\sigma_s$*, public keys* $\mathsf{ipk}, \mathsf{opk}$*, and a message* $m \in \{0,1\}^*$*, output a group signature* $\sigma_m$*:*

$$\sigma_m \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m).$$

$\mathsf{Vfy}_\Delta$. *On input* $\mathsf{ipk}, \mathsf{opk}, m, \sigma_m$*, output a decision bit:*

$$d \leftarrow \mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m).$$

$\mathsf{Opn}_\Delta$. *On input the opener secret key* $\mathsf{osk}$*, the issuer public key* $\mathsf{ipk}$*, a message* $m$*, and a signature* $\sigma_m$*, output a member public key* $\mathsf{upk}$ *and opening information* $O$*:*

$$(\mathsf{upk}, O) \leftarrow_\$ \mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m).$$

$\mathsf{Jdg}_\Delta$. *On input* $\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}, O$*, output a decision bit:*

$$d \leftarrow \mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}, O).$$

**Definition 5 (Correctness).** $\Delta$ *is correct if for all* $\lambda \in \mathbb{N}$*, all* $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$*, all key pairs* $(\mathsf{isk}, \mathsf{ipk}) \leftarrow_\$ \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$*,* $(\mathsf{osk}, \mathsf{opk}) \leftarrow_\$ \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$*,* $(\mathsf{usk}, \mathsf{upk}) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$*, all* $\sigma_s \leftarrow_\$ \mathsf{Join}_\Delta(\mathsf{isk}, \mathsf{upk}, \mathsf{opk})$*, all* $m \in \mathcal{MS}$ *(normally* $\{0,1\}^*$*), and all* $\sigma_m \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)$*, we have* $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m) = 1$ *and for all* $(\mathsf{upk}', O) \leftarrow_\$ \mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m) : \mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}', O) = 1$.

**Security.** A secure group signature scheme must satisfy the following fundamental properties: *anonymity*, *traceability*, *non-frameability*, *trace-soundness*, and *non-claimability*. These *almost* fully capture the required behavior, i.e., users enjoy anonymity, but misbehavior can be traced when required. We already incorporate our strengthened definitions for traceability, trace-soundness, and non-claimability.

1. **Anonymity:** Signatures do not reveal the signer's identity. Even if the adversary controls the issuer, and knows all the users' secrets, it should not distinguish which of two honest users produced a given signature—unless it queries the opener.

2. **(Strengthened) Traceability:** Every valid signature must be openable by the prescribed algorithm, even if the opener's *secret key* is leaked. Our strengthening also accounts for a corrupt issuer.
3. **Non-frameability:** No adversary can produce a valid signature that (under its chosen public context) opens to an honest user who did not create it—even if the issuer and opener are corrupt.
4. **(Strengthened) Trace-soundness:** No single signature admits two *different* valid openings to different users. We strengthen this property to also cover groups.
5. **(Strengthened) Non-claimability:** A signature generated by an honest member cannot be attributed to another user. In our strengthened definition, the adversary also receives the user's secret key.

*Remark 1.* In existing definitions [5,10], the creations of members - and corruption thereof - are modeled via oracles. We opted to use fixed honest members. These flavors are polynomially equivalent, i.e., the adversary can simulate honest users itself. Hence, a reduction thus needs to guess which users will not be corrupted. However, our model seems to be more intuitive.

**Anonymity.** Figure 4 formalizes anonymity against a malicious issuer while assuming an honest opener. The challenger generates public parameters $\mathsf{pp}_\Delta$, the opener's key pair $(\mathsf{osk}, \mathsf{opk})$, and two honest members $(\mathsf{usk}_0, \mathsf{upk}_0)$, $(\mathsf{usk}_1, \mathsf{upk}_1)$. A hidden bit $b \in \{0, 1\}$ determines which member is used in challenge signatures. The adversary $\mathcal{A}$ receives $\mathsf{opk}$ and both member secret keys and interacts with two oracles:

- *Left-or-right signing oracle* $\mathcal{O}^{b,\mathsf{usk}_0,\mathsf{usk}_1}_{\mathsf{LoRSign}}$ takes $(m, \mathsf{ipk}, \sigma_s^0, \sigma_s^1)$, produces two candidate signatures under the respective members, and returns only the $b$-th one $\sigma_m^b$. Both candidates are verified first, ensuring that malformed inputs cannot trivially leak information. The tuple $(\sigma_m^b, m, \mathsf{ipk})$ is added to a set $\mathcal{Q}_\sigma$.
- *Opening oracle* $\mathcal{O}^{\mathsf{osk}}_{\mathsf{Opn}'_\Delta}$ takes $(\mathsf{ipk}, m, \sigma_m)$, but refuses to open if $(\sigma_m, m, \mathsf{ipk}) \in \mathcal{Q}_\sigma$, i.e., if the signature is a challenge transcript. Otherwise, it runs $\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m)$. This exclusion rule prevents the opener from trivially de-anonymizing the challenge signatures.

Finally, $\mathcal{A}$ outputs a bit $b^*$. The experiment returns 1 iff $b^* = b$. Anonymity requires that no PPT adversary can guess $b$ with advantage non-negligibly better than $1/2$, even though it controls issuer contexts and certificates, and has full access to the honest opener—except on the explicitly excluded challenge signatures.

**Definition 6 (Anonymity).** $\Delta$ *satisfies anonymity if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\nu$ *such that*

$$\left|\Pr\left[\mathbf{Exp}^{\mathsf{Anon}}_{\mathcal{A},\Delta}(\lambda) = 1\right] - {}^1\!/_2\right| \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 4.

```
$\mathbf{Exp}^{\mathsf{Anon}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $(\mathsf{osk}, \mathsf{opk}) \leftarrow_\$ \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$
  $(\mathsf{usk}_0, \mathsf{upk}_0) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$
  $(\mathsf{usk}_1, \mathsf{upk}_1) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$
  $b \leftarrow_\$ \{0, 1\}$
  $\mathcal{Q}_\sigma \leftarrow \emptyset$
  $b^* \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{b,\mathsf{usk}_0,\mathsf{usk}_1}_{\mathsf{LoRSign}}, \mathcal{O}^{\mathsf{osk}}_{\mathsf{Opn}'_\Delta}}(\mathsf{opk}, \mathsf{usk}_0, \mathsf{usk}_1)$
    where $\mathcal{O}^{b,\mathsf{usk}_0,\mathsf{usk}_1}_{\mathsf{LoRSign}}(m, \mathsf{ipk}, \sigma_s^0, \sigma_s^1)$:
      $\sigma_m^0 \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}_0, \sigma_s^0, \mathsf{ipk}, \mathsf{opk}, m)$
      $\sigma_m^1 \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}_1, \sigma_s^1, \mathsf{ipk}, \mathsf{opk}, m)$
      return $\perp$ if $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m^0) \neq 1 \ \vee \ \mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m^1) \neq 1$
      $\mathcal{Q}_\sigma \leftarrow \mathcal{Q}_\sigma \cup \{(\sigma_m^b, m, \mathsf{ipk})\}$
      return $\sigma_m^b$
    where $\mathcal{O}^{\mathsf{osk}}_{\mathsf{Opn}'_\Delta}(\mathsf{ipk}, m, \sigma_m)$:
      return $\perp$ if $(\sigma_m, m, \mathsf{ipk}) \in \mathcal{Q}_\sigma$
      return $\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m)$
  return $1$ if $b^* = b$; else $0$
```

Fig. 4: Anonymity of group signature $\Delta$

**Traceability.** Figure 5 formalizes traceability in the presence of a malicious issuer. The challenger generates public parameters $\mathsf{pp}_\Delta$ and an honest opener key pair $(\mathsf{osk}, \mathsf{opk})$. The adversary $\mathcal{A}$ then outputs an issuer public key $\mathsf{ipk}^*$, a message $m^*$, and a candidate group signature $\sigma_m^*$. Thus, the adversary controls the issuer and all users.

The challenger first checks whether $\sigma_m^*$ verifies under $(\mathsf{ipk}^*, \mathsf{opk})$. If verification fails, the adversary loses immediately. Otherwise, the honest opener uses $\mathsf{osk}$ to open the signature, producing a purported member public key $\mathsf{upk}^*$ and opening proof $O$. The challenger finally tests the validity of this opening via $\mathsf{Jdg}_\Delta$. If the opening procedure does not yield a valid and consistent result, the adversary has succeeded in producing an *untraceable* signature, and the experiment returns $1$.

Traceability requires that no PPT adversary can win this game with non-negligible probability: every valid signature must admit a verifiable opening with respect to the honest opener, even when the issuer is malicious.

**Definition 7 (Traceability).** $\Delta$ *satisfies traceability if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\nu$ *such that*

$$\Pr\Big[\mathbf{Exp}^{\mathsf{Traceability}}_{\mathcal{A},\Delta}(\lambda) = 1\Big] \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 5.

```
$\mathbf{Exp}^{\mathsf{Traceability}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $(\mathsf{osk}, \mathsf{opk}) \leftarrow_\$ \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$
  $(\mathsf{ipk}^*, m^*, \sigma_m^*) \leftarrow_\$ \mathcal{A}(\mathsf{osk})$
  return $0$ if $\mathsf{Vfy}_\Delta(\mathsf{ipk}^*, \mathsf{opk}, m^*, \sigma_m^*) \neq 1$
  $(\mathsf{upk}^*, O) \leftarrow_\$ \mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}^*, m^*, \sigma_m^*)$
  return $1$ if $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*, \mathsf{opk}, m^*, \sigma_m^*, \mathsf{upk}^*, O) \neq 1$
  return $0$
```

Fig. 5: Traceability of group signature $\Delta$

**Non-Frameability.** Figure 6 defines non-frameability. The challenger generates public parameters $\mathsf{pp}_\Delta$ and fixes one honest user with key pair $(\mathsf{usk}, \mathsf{upk})$. The adversary $\mathcal{A}$ receives $\mathsf{upk}$ and can adaptively query a signing oracle $\mathcal{O}^{\mathsf{usk}}_{\mathsf{Sgn}_\Delta}$ to obtain signatures from the honest user on messages of its choice under contexts $(\mathsf{ipk}, \mathsf{opk})$ and a membership certificate $\sigma_s$. Each oracle output is recorded in the set $\mathcal{Q}$. At the end, the adversary outputs $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, O^*)$. If the submitted tuple corresponds to an honest-user signature already in $\mathcal{Q}$, the attempt is rejected. Otherwise, the challenger runs $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, \mathsf{upk}, O^*)$. If this verification succeeds, the adversary has *framed* the honest user by producing a signature and opening that implicates her without actually having created it. Non-frameability therefore guarantees that no PPT adversary can cause an honest member to be falsely accused of generating a signature it never produced, even when the issuer, the opener, and all users collude.

```
$\mathbf{Exp}^{\mathsf{NonFrame}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $(\mathsf{usk}, \mathsf{upk}) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$
  $\mathcal{Q}_\sigma \leftarrow \emptyset$
  $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, O^*) \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{\mathsf{usk}}_{\mathsf{Sgn}_\Delta}}(\mathsf{upk})$
    where $\mathcal{O}^{\mathsf{usk}}_{\mathsf{Sgn}_\Delta}(\mathsf{ipk}, \mathsf{opk}, \sigma_s, m)$:
      $\sigma_m \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)$
      $\mathcal{Q}_\sigma \leftarrow \mathcal{Q}_\sigma \cup \{(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m)\}$
      return $\sigma_m$
  return $0$ if $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*) \in \mathcal{Q}_\sigma$
  return $1$ if $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, \mathsf{upk}, O^*) = 1$
  return $0$
```

Fig. 6: Non-Frameability of group signature $\Delta$

**Definition 8 (Non-Frameability).** $\Delta$ *is non-frameable if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\nu$ *such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{NonFrame}}_{\mathcal{A},\Delta}(\lambda) = 1\right] \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 6.

**Trace-Soundness.** Figure 7 captures trace-soundness. The challenger generates public parameters $\mathsf{pp}_\Delta$ and gives them to the adversary. The adversary then outputs a single message $m^*$, a candidate group signature $\sigma^*_m$, and two purported openings $(\mathsf{ipk}^*_0, \mathsf{opk}^*_0, \mathsf{upk}^*_0, O^*_0)$ and $(\mathsf{ipk}^*_1, \mathsf{opk}^*_1, \mathsf{upk}^*_1, O^*_1)$.

The experiment rejects immediately if both openings are identical in all three components $(\mathsf{upk}, \mathsf{ipk}, \mathsf{opk})$. It also rejects if either of the two openings fails verification under $\mathsf{Jdg}_\Delta$. The adversary wins in all other cases.

Trace-soundness ensures that no valid group signature can be opened in two conflicting ways. In other words, each message–signature pair admits at most one valid opening, thereby enforcing consistency and precluding equivocation even across different groups.

```
$\mathbf{Exp}^{\mathsf{TraceSound}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $(\mathsf{ipk}^*_0, \mathsf{opk}^*_0, \mathsf{ipk}^*_1, \mathsf{opk}^*_1, m^*, \sigma^*_m, \mathsf{upk}^*_0, O^*_0, \mathsf{upk}^*_1, O^*_1) \leftarrow_\$ \mathcal{A}(\mathsf{pp}_\Delta)$
  return 0 if $(\mathsf{upk}^*_0, \mathsf{ipk}^*_0, \mathsf{opk}^*_0) = (\mathsf{upk}^*_1, \mathsf{ipk}^*_1, \mathsf{opk}^*_1)$
  return 0 if $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*_0, \mathsf{opk}^*_0, m^*, \sigma^*_m, \mathsf{upk}^*_0, O^*_0) \neq 1$
  return 0 if $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*_1, \mathsf{opk}^*_1, m^*, \sigma^*_m, \mathsf{upk}^*_1, O^*_1) \neq 1$
  return 1
```

Fig. 7: Trace-Soundness of group signature $\Delta$

**Definition 9 (Trace-Soundness).** $\Delta$ *is trace-sound if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\nu$ *such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{TraceSound}}_{\mathcal{A},\Delta}(\lambda) = 1\right] \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 7.

**Non-Claimability.** Figure 8 defines non-claimability. The challenger generates public parameters $\mathsf{pp}_\Delta$ and fixes one honest user with key pair $(\mathsf{usk}, \mathsf{upk})$. In our version, the adversary $\mathcal{A}$ receives the user's *secret* key $\mathsf{usk}$ and can adaptively query a signing oracle $\mathcal{O}^{\mathsf{usk}}_{\mathsf{Sgn}_\Delta}$ to obtain valid signatures from the honest user under contexts of its choice $(\mathsf{ipk}, \mathsf{opk}, \sigma_s, m)$. Each oracle response is logged in the set $\mathcal{Q}$.

At the end, $\mathcal{A}$ outputs $(\mathsf{ipk}^*, \mathsf{opk}^*, \mathsf{upk}^*, m^*, \sigma^*_m, O^*)$. The experiment checks that $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m)$ was indeed one of the signatures produced by the honest user (so only *re-attribution* is tested), and rejects if the adversary tries to point back to the same user ($\mathsf{upk} = \mathsf{upk}^*$). The adversary wins if it nevertheless produces an accepting opening that implicates a *different* $\mathsf{upk}^*$.

In other words, non-claimability ensures that once a signature has been honestly produced by a fixed honest member, no adversary—no matter what public context it chooses—can later "re-assign" that signature to a different member with a valid opening. This notion was first introduced by Bootle et al. [10] (under the name *opening-soundness*); we retain it as a distinct property to avoid confusion with the different notion of trace-soundness introduced by Sakai et al. [23].

```
$\mathbf{Exp}^{\mathsf{NonClaimability}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $(\mathsf{usk}, \mathsf{upk}) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$
  $\mathcal{Q}_\sigma \leftarrow \emptyset$
  $(\mathsf{ipk}^*, \mathsf{opk}^*, \mathsf{upk}^*, m^*, \sigma^*_m, O^*) \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{\mathsf{usk}}_{\mathsf{Sgn}_\Delta}}(\mathsf{usk})$
    where $\mathcal{O}^{\mathsf{usk}}_{\mathsf{Sgn}_\Delta}(\mathsf{ipk}, \mathsf{opk}, \sigma_s, m)$:
      $\sigma_m \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)$
      $\mathcal{Q}_\sigma \leftarrow \mathcal{Q}_\sigma \cup \{(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m)\}$
      return $\sigma_m$
  return $0$ if $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m) \notin \mathcal{Q}_\sigma$
  return $0$ if $\mathsf{upk} = \mathsf{upk}^*$
  return $1$ if $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m, \mathsf{upk}^*, O^*) = 1$
  return $0$
```

Fig. 8: Non-Claimability of group signature $\Delta$

**Definition 10 (Non-Claimability).** *$\Delta$ is non-claimable if for every PPT adversary $\mathcal{A}$ there exists a negligible function $\nu$ such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{NonClaimability}}_{\mathcal{A},\Delta}(\lambda) = 1\right] \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 8.

## 4 A Secure Yet Forgeable Instantiation

We now present a deliberately contrived construction, included purely as an illustrative example. The construction is kept as simple as possible to make the definitional gap evident. Thus, its role is not to introduce a new cryptographic primitive, but to provide a minimal counterexample: even when all BSZ-style security notions (even in our already strengthened formulation) are formally satisfied, a corrupted opener that generates its own keys can still undermine unforgeability by minting signatures and fabricating openings. This shows that the traditional BSZ properties, even with its extensions, while coherent in their own right, remain insufficient in adversarial settings.

**Overview.** Our construction formally satisfies all standard security properties, even in their strengthened form, yet explicitly allows a corrupted opener to violate unforgeability and generate arbitrary openings. This is achieved through a *dual-mode verification mechanism*: a signature is accepted either via a NIZK proof or via a digital signature under a special key embedded into the opener's public key. This secondary verification path is only available to a malicious opener.

**Construction Rationale.** The scheme follows a canonical group signature structure. Namely, users receive membership certificates from the issuer, the opener possesses an encryption key for tracing identities, and signers prove (via a NIZK) that they hold a valid certificate and identity preimage.

However, in the presence of an adversary controlling the opener from the beginning, two distinct attack vectors emerge:

1. **Forged Signatures without Join.** During setup, the opener generates an auxiliary signing key pair $(\mathsf{sk}'_\Sigma, \mathsf{pk}'_\Sigma)$, where $\mathsf{pk}'_\Sigma$ becomes part of the public opener key. A verifier accepts a signature either if the NIZK proof verifies, or if $\sigma'$ is a valid signature under $\mathsf{pk}'_\Sigma$. A malicious opener who retains $\mathsf{sk}'_\Sigma$ can therefore forge signatures by directly signing under this key—without ever running the join protocol, and without any issuer interaction or secret identity $w$. These forgeries form valid group signatures but cannot be traced to any honest user.
2. **Bogus Openings for Arbitrary Identities.** In addition, the malicious opener can embed arbitrary values $y^* = f(w^*)$ of its choice into $\sigma'$ and construct an opening that points to these identities. This allows producing valid-looking openings for arbitrary users that never joined. However, due to the one-wayness of $f$, the opener *cannot* produce such an opening for any honest user. Thus, while the system permits openings for unregistered users, it still preserves *non-frameability* for all honest members and, somewhat surprisingly, also *trace-soundness* and *non-claimability*.

This construction illustrates that existing definitions leave room for malicious behavior if the opener is corrupted. This underscores a key insight: traditional security properties are insufficient to guarantee what *intuitively* is expected from group signatures.

**Languages.** We define two NP languages used in the construction:

$$L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta} := \left\{ (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, c, m) \;\middle|\; \begin{aligned} &\exists (r, \sigma_s, w) : \\ &\mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, f(w)), \sigma_s) = 1 \\ &\wedge\ c = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, f(w); r) \end{aligned} \right\}$$

$$L_2^{f,\mathsf{pp}_\Theta} := \left\{ (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, c, m, y) \;\middle|\; \begin{aligned} &\exists (\mathsf{sk}_\Theta) : y \leftarrow \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c) \\ &\wedge\ \mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1 \end{aligned} \right\}$$

**Construction.** Construction 2 specifies the scheme. The setup algorithm $\mathsf{PG}_\Delta$ generates all necessary public parameters, including those for the digital signature scheme $\Sigma$, the encryption scheme $\Theta$, and the NIZK systems. Key generation for the member, issuer, and opener follows the expected roles.

The join process is simple: The issuer generates a certificate consisting of all relevant public keys and the member's identity $y = f(w)$. This certificate is later referenced in the signing algorithm, where the member proves possession of a valid certificate and that $c$ is an encryption of the same identity.

The signing algorithm encrypts the member identity $f(w)$ into a ciphertext $c$ and generates a NIZK proof $\pi$ for membership in $L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta}$, using as witness the certificate, encryption randomness, and the secret $w$.

The verification algorithm accepts two cases:

1. Either the signature $(c, \pi, \sigma', y')$ passes NIZK verification for $L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta}$ (with $\sigma' = \perp$ and $y' = \perp$), or
2. $\sigma'$ is itself a signature under the auxiliary verification key $\mathsf{pk}'_\Sigma$.

The second case captures the corrupted opener's ability to forge. In all other aspects, the scheme behaves like a standard group signature: honest signatures are unlinkable, opening is possible, and framing is prevented by the NIZK.

Trace-soundness is ensured because the ciphertext $c$ is binding and uniquely decrypts to $y$; the NIZK proof $\pi_o$ ensures that the opening is valid and consistent.

This construction thus demonstrates a model where non-frameability and trace-soundness hold in the presence of a corrupted opener, while unforgeability breaks in a controlled and analyzable way.

One additional important detail is that $\mathsf{Jdg}_\Delta$ returns 0 if an "obvious" forgery under the opener's public key is seen.

**Theorem 8.** *Let $\Delta$ be the scheme from Construction 2. Then $\Delta$ is perfectly correct.*

*Proof.* Follows by construction. □

```
$\underline{\mathsf{PG}_\Delta(1^\lambda)}$:
  – Choose a one-way function $f$,
  – $\mathsf{pp}_\Sigma \leftarrow_\$ \mathsf{PP}_\Sigma(1^\lambda)$,
  – $\mathsf{pp}_\Theta \leftarrow_\$ \mathsf{PP}_\Theta(1^\lambda)$,
  – $\mathsf{crs}^1_\Pi \leftarrow_\$ \mathsf{PG}_\Pi(1^\lambda)$ for $L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta}$,
  – $\mathsf{crs}^2_\Pi \leftarrow_\$ \mathsf{PG}_\Pi(1^\lambda)$ for $L_2^{f,\mathsf{pp}_\Theta}$.
  – Return
    $\mathsf{pp}_\Delta = (\mathsf{crs}^1_\Pi, \mathsf{crs}^2_\Pi, \mathsf{pp}_\Sigma, \mathsf{pp}_\Theta, f)$.
$\underline{\mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)}$:
  – Parse
    $\mathsf{pp}_\Delta = (\mathsf{crs}^1_\Pi, \mathsf{crs}^2_\Pi, \mathsf{pp}_\Sigma, \mathsf{pp}_\Theta, f)$,
  – Sample $w \leftarrow_\$ C_\lambda$,
  – Set $y \leftarrow f(w)$,
  – Return $(\mathsf{usk}, \mathsf{upk}) = (w, y)$.
$\underline{\mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)}$:
  – Parse $\mathsf{pp}_\Delta$ as before,
  – $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_\$ \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma)$,
  – Return $(\mathsf{isk}, \mathsf{ipk}) = (\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma)$.
$\underline{\mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)}$:
  – Parse $\mathsf{pp}_\Delta$ as before.
  – $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) \leftarrow_\$ \mathsf{KG}_\Theta(\mathsf{pp}_\Theta)$,
  – $\boxed{(\mathsf{sk}'_\Sigma, \mathsf{pk}'_\Sigma) \leftarrow_\$ \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma)}$,
  – Return $(\mathsf{osk}, \mathsf{opk}) = (\mathsf{sk}_\Theta, (\mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma}))$.
$\underline{\mathsf{Join}_\Delta(\mathsf{isk}, \mathsf{upk}, \mathsf{opk})}$:
  – Parse $\mathsf{sk}_\Sigma = \mathsf{isk}$, $\mathsf{upk} = y$, $\mathsf{opk} = (\mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma})$,
  – Return $\mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma}, y))$.
$\underline{\mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)}$:
  – Parse $\mathsf{usk} = w$, $\mathsf{upk} = y = f(w)$, $\mathsf{ipk} = \mathsf{pk}_\Sigma$, $\mathsf{opk} = (\mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma})$,
  – $(c; r) \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, f(w))$,
  – $\pi \leftarrow_\$ \mathsf{Prf}_\Pi(\mathsf{crs}^1_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma}, c, m), (r, \sigma_s, w))$ for $L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta}$,
  – Return $\sigma_m = (c, \pi \boxed{, \bot, \bot})$.
$\underline{\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m)}$:
  – Parse $(c, \pi \boxed{, \sigma', y'}) = \sigma_m$, $\mathsf{opk} = (\mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma})$, $\mathsf{ipk} = \mathsf{pk}_\Sigma$,
  – If $\mathsf{Vfy}_\Sigma(\mathsf{pk}'_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma}, y', m, c, \pi), \sigma') = 1$ return $1$,
  – Return $\mathsf{Vfy}_\Pi(\mathsf{crs}^1_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma}, c, m), \pi)$.
$\underline{\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m)}$:
  – If $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m) \neq 1$ return $0$,
  – Parse $\mathsf{osk} = \mathsf{sk}_\Theta$, $\sigma_m = (c, \pi \boxed{, \sigma', y'})$, $\mathsf{ipk} = \mathsf{pk}_\Sigma$, $\mathsf{opk} = (\mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma})$,
  – Decrypt $y \leftarrow \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c)$,
  – Generate $\pi_o \leftarrow_\$ \mathsf{Prf}_\Pi(\mathsf{crs}^2_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma}, c, m, y), (\mathsf{sk}_\Theta))$ for $L_2^{f,\mathsf{pp}_\Theta}$,
  – Return $(\mathsf{upk}, O) = (y, \pi_o)$.
$\underline{\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}, O)}$:
  – If $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m) \neq 1$ return $0$,
  – Parse $O = \pi_o$, $\mathsf{upk} = y$, $\mathsf{opk} = (\mathsf{pk}_\Theta \boxed{, \mathsf{pk}'_\Sigma})$, $(c, \pi \boxed{, \sigma', y'}) = \sigma_m$, $\mathsf{ipk} = \mathsf{pk}_\Sigma$,
  – $\boxed{\text{Return 1 if } y' = f(\pi_o) = f(O) = \mathsf{upk}}$,
  – $\boxed{\text{Return 0 if } \sigma' \neq \bot \vee y' \neq \bot,}$
  – Return $\mathsf{Vfy}_\Pi(\mathsf{crs}^2_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, c, m, y), \pi_o)$.
```

Construction 2: Contrived Group Signature Scheme $\Delta$. The additional bypass for the opener is marked with a $\boxed{\text{box}}$. This bypass is dropped for the final construction.

**Security.** We prove each property separately. An overview of the used assumptions is given in Table 3. For anonymity we also use traceability.

**Theorem 9 (Trace-Soundness).** *$\Delta$ is trace-sound.*

*Intuition.* The proof follows a sequence of games. Essentially, ZK, SSE, context-binding or key-verifiability must break, if the adversary wins.

| **Property** | **ZK** | **SSE** | **SB** | **KV** | **OWF** | **$\Sigma$-UNF** | **DS** | **$\Theta$-CPA** | **Trace** |
|---|---|---|---|---|---|---|---|---|---|
| Anonymity | 2 | $q_0$ | | | | | | 1 | 1 |
| Non-frameability | 2 | 2 | | | 2 | | 1 | | |
| Traceability | 1 | 1 | | | | 1 | | | |
| Trace-soundness | 2 | 4 | 1 | 1 | | | | | |
| Non-claimability | 1 | 1 | | | | | 1 | | |

Table 3: Assumptions required for each security property of the contrived construction $\Delta$. Entries are the corresponding loss; an empty cell means the assumption is not required. The parameter $q_0$ denotes the adversary's query bound for the relevant game (as defined in the text).

The proof is given in Appendix C.1.

**Theorem 10 (Non-Frameability).** $\Delta$ *is non-frameable.*

*Intuition.* The proof follows a sequence of games. Essentially ZK, SSE, decryption-soundness or onewayness cannot hold, if the adversary wins.

The proof is given in Appendix C.2.

**Theorem 11 (Traceability).** $\Delta$ *is traceable.*

*Intuition.* The proof follows a sequence of games. Essentially ZK, SSE, unforgeability of $\Sigma$ (the bypass signature) cannot hold, if the adversary wins.

The proof is given in Appendix C.3.

**Theorem 12 (Anonymity).** $\Delta$ *is anonymous.*

*Intuition.* Essentially, ZK, SSE, or the mcIND-CPA of $\Theta$ cannot hold, if the adversary wins.

The proof is given in Appendix C.4.

**Theorem 13 (Non-Claimability).** $\Delta$ *is non-claimable.*

*Intuition.* Essentially ZK, SSE, or decryption-soundness cannot hold, if the adversary wins.

The proof is given in Appendix C.5.

**How to Sign as a Corrupt Opener.** The contrived construction exploits the fact that *anonymity and traceability are irrelevant* once the opener is corrupted. Instead, the corrupted opener can leverage the *auxiliary verification branch* in $\mathsf{Vfy}_{\Delta}$: recall that verification accepts either if the NIZK proof $\pi$ verifies, or if the signature contains a valid $\Sigma$ signature $\sigma'$ under the auxiliary key $\mathsf{pk}'_{\Sigma}$ (embedded in the opener's public key).

While an honest opener discards $\mathsf{sk}'_\Sigma$ at setup, a malicious opener retains it and can directly sign arbitrary tuples

$$(\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, y', m, c, \pi).$$

The resulting signature $\sigma_m = (\cdot, \cdot, \sigma', y')$ is then accepted by any verifier via the auxiliary branch, even though no group member and no witness $w$ were involved. This yields the intended separation: the scheme remains non-frameable and trace-sound, yet a malicious opener can *forge* group signatures at will.

**How to Create Bogus Openings.** A malicious opener can also fabricate *plausible openings* for such forged signatures. Under $\mathsf{Jdg}_\Delta$, if the auxiliary branch verifies $(\sigma', y')$, the verifier discards decryption and accepts an opening whenever the opener provides a preimage $w$ such that $f(w) = y'$.

Hence, without ever decrypting $c$, the opener can produce accepting openings to *arbitrary, non-joined identities* of its choice by simply revealing self-generated preimages. Crucially, the opener *cannot* redirect a signature to an honest user's identity $y = f(w_{\mathsf{hon}})$, since this would require revealing $w_{\mathsf{hon}}$, contradicting one-wayness of $f$. Therefore, non-frameability and non-claimability are preserved, even though the opener can manufacture bogus openings.

## 5 Closing the Gap: Security Against Malicious Openers

Our contrived construction shows that group signatures require unforgeability guarantees beyond correctness and traceability. The system must reject (i) signatures and (ii) openings that are not tied to legitimately joined members. On the signature side, we use two experiments, reflecting that a valid group signature depends on both an honestly issued membership certificate and the corresponding member secret key: *certificate-unforgeability* rules out signing without a legitimate certificate, and *key-unforgeability* rules out signing without a registered member key under the honest issuer.

**Certificate-Unforgeability.** We model an adversary that fully controls the opener, while the issuer and members are honest. The adversary receives the global parameters and accesses three oracles: (1) member key generation (returns fresh $(\mathsf{usk}, \mathsf{upk})$), (2) join (runs the honest registration with the issuer and returns a handle $i$ to the resulting certificate), and (3) signing (produces a group signature under an honestly issued certificate referenced by $i$). The adversary outputs $(\sigma^*_m, m^*, \mathsf{opk}^*)$. The experiment rejects if this is a replay of a previously obtained signature and otherwise accepts if $\sigma^*_m$ verifies under the honest issuer key $\mathsf{ipk}$.

Intuition: the opener cannot mint certificates or sign on behalf of non-members. The property is meaningful only against a *malicious opener* with an *honest issuer*; if the issuer were adversarial, it could simulate all joins and signatures on its own using the member secret keys it learns.

```
$\mathbf{Exp}^{\mathsf{CertUnforgeability}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $\mathcal{Q}_V \leftarrow \emptyset$ // member secret/public pairs
  $\mathcal{Q}_\sigma \leftarrow \emptyset$ // seen signatures (for replay)
  $\boxed{\mathcal{Q}_{\mathsf{cert}} \leftarrow \emptyset}$ // certificates
  $\boxed{i \leftarrow 0}$ // index
  $(\mathsf{isk}, \mathsf{ipk}) \leftarrow_\$ \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$
  $(\sigma_m^*, m^*, \mathsf{opk}^*) \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\mathsf{KgM}_\Delta}, \mathcal{O}_{\mathsf{Sgn}_\Delta}, \mathcal{O}^{\mathsf{isk}}_{\mathsf{Join}_\Delta}}(\mathsf{ipk})$
    where oracle $\mathcal{O}_{\mathsf{KgM}_\Delta}()$:
      $(\mathsf{usk}, \mathsf{upk}) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$
      $\mathcal{Q}_V \leftarrow \mathcal{Q}_V \cup \{(\mathsf{usk}, \mathsf{upk})\}$;
      return $\boxed{(\mathsf{usk}, \mathsf{upk})}$
    where oracle $\boxed{\mathcal{O}_{\mathsf{Sgn}_\Delta}(i, \mathsf{opk}, m):}$
      return $\perp$ if no entry of the form
        $\boxed{(i, \sigma_s, \mathsf{usk}, \mathsf{upk}) \text{ exists in } \mathcal{Q}_{\mathsf{cert}}}$
      $\boxed{\sigma_m \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)}$
      $\boxed{\mathcal{Q}_\sigma \leftarrow \mathcal{Q}_\sigma \cup \{(\sigma_m, m, \mathsf{opk})\}}$
      return $\sigma_m$
    where oracle $\mathcal{O}^{\mathsf{isk}}_{\mathsf{Join}_\Delta}(\mathsf{opk}, \mathsf{upk})$:
      return $\perp$ if no entry of the form
        $(\mathsf{usk}, \mathsf{upk})$ exists in $\mathcal{Q}_V$
      $\sigma_s \leftarrow_\$ \mathsf{Join}_\Delta(\mathsf{isk}, \mathsf{upk}, \mathsf{opk})$
      $\boxed{i \leftarrow i + 1}$
      $\boxed{\mathcal{Q}_{\mathsf{cert}} \leftarrow \mathcal{Q}_{\mathsf{cert}} \cup \{(i, \sigma_s, \mathsf{usk}, \mathsf{upk})\}}$
      return $i$
  return 0 if $\boxed{(\sigma_m^*, m^*, \mathsf{opk}^*) \in \mathcal{Q}_\sigma}$
  return 1 if $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}^*, m^*, \sigma_m^*) = 1$
  return 0
```

(a) Certificate-Unforgeability of group signature Δ

```
$\mathbf{Exp}^{\mathsf{KeyUnforgeability}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $\mathcal{Q}_V \leftarrow \emptyset$ // member secret/public pairs
  $\mathcal{Q}_\sigma \leftarrow \emptyset$ // seen signatures (for replay)
  $\boxed{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxx}}$
  $\boxed{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxx}}$
  $(\mathsf{isk}, \mathsf{ipk}) \leftarrow_\$ \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$
  $(\sigma_m^*, m^*, \mathsf{opk}^*) \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\mathsf{KgM}_\Delta}, \mathcal{O}_{\mathsf{Sgn}_\Delta}, \mathcal{O}^{\mathsf{isk}}_{\mathsf{Join}_\Delta}}(\mathsf{ipk})$
    where oracle $\mathcal{O}_{\mathsf{KgM}_\Delta}()$:
      $(\mathsf{usk}, \mathsf{upk}) \leftarrow_\$ \mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$
      $\mathcal{Q}_V \leftarrow \mathcal{Q}_V \cup \{(\mathsf{usk}, \mathsf{upk})\}$
      return $\boxed{\mathsf{upk}}$
    where oracle $\mathcal{O}_{\mathsf{Sgn}_\Delta}(\boxed{\mathsf{ipk}', \mathsf{upk}, \sigma_s}, \mathsf{opk}, m)$:
      return $\perp$ if no entry of the form
        $\boxed{(\mathsf{usk}, \mathsf{upk}) \in \mathcal{Q}_V \text{ exists in } \mathcal{Q}_V}$
      $\sigma_m \leftarrow_\$ \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \boxed{\mathsf{ipk}'}, \mathsf{opk}, m)$
      $\mathcal{Q}_\sigma \leftarrow \mathcal{Q}_\sigma \cup \{(\sigma_m, m, \mathsf{opk}\boxed{, \mathsf{ipk}'})\}$
      return $\sigma_m$
    where oracle $\mathcal{O}^{\mathsf{isk}}_{\mathsf{Join}_\Delta}(\mathsf{opk}, \mathsf{upk})$:
      return $\perp$ if no entry of the form
        $(\mathsf{usk}, \mathsf{upk})$ exists in $\mathcal{Q}_V$
      $\sigma_s \leftarrow_\$ \mathsf{Join}_\Delta(\mathsf{isk}, \mathsf{upk}, \mathsf{opk})$
      $\boxed{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxx}}$
      $\boxed{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxx}}$
      return $\sigma_s$
  return 0 if $(\sigma_m^*, m^*, \mathsf{opk}^*\boxed{, \mathsf{ipk}}) \in \mathcal{Q}_\sigma$
  return 1 if $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}^*, m^*, \sigma_m^*) = 1$
  return 0
```

(b) Key-Unforgeability of group signature Δ

Fig. 9: Side-by-side view of the two experiments. Left: certificate-unforgeability uses a certificate index $i$ and a certificate set $\mathcal{Q}_{\mathsf{cert}}$ to gate signing. Right: key-unforgeability dispenses with certificate handles and instead tracks registered member keys and the issuer key inside the signing context; the replay filter includes the issuer key, as in the certificate-unforgeability experiment, the adversary can simulate issuing for other issuer keys itself.

**Definition 11 (Certificate-Unforgeability).** *A group signature scheme Δ offers certificate-unforgeability if for every PPT adversary $\mathcal{A}$,*

$$\Pr\left[\mathbf{Exp}^{\mathsf{CertUnforgeability}}_{\mathcal{A},\Delta}(\lambda) = 1\right] \leq \nu(\lambda).$$

See Figure 9a for the experiment.

**Key-Unforgeability.** Here the adversary again acts as a malicious opener against an honest issuer and honest members. It may register arbitrary users via the join oracle. The signing oracle takes as input the member key and an *issuer key* and returns a valid signature for that context. At the end, the adversary outputs $(\sigma_m^*, m^*, \mathsf{opk}^*)$. The experiment accepts if $\sigma_m^*$ verifies under the honest issuer key $\mathsf{ipk}$ and is not a replay. The replay filter treats the issuer key as part of the signature context.

Intuition: even if the opener can register arbitrary users, it cannot produce a fresh valid signature under the honest issuer's key without a legitimately registered member key. The explicit issuer-key parameter prevents "issuer swapping" attacks and aligns the replay check with the exact signing context.

**Definition 12 (Key-Unforgeability).** *A group signature scheme $\Delta$ offers key-unforgeability if for every PPT adversary $\mathcal{A}$,*

$$\Pr\left[\mathbf{Exp}_{\mathcal{A},\Delta}^{\mathsf{KeyUnforgeability}}(\lambda) = 1\right] \leq \nu(\lambda).$$

See Figure 9b for the experiment.

**Opening Unforgeability.** Opening unforgeability requires that only joined identities can be pinpointed by an accepting opening. Even a powerful adversary who interacts with an honest issuer and obtains joining transcripts for arbitrarily many pairs $(\mathsf{opk}, \mathsf{upk})$ must not be able to manufacture a new tuple $(\mathsf{opk}^*, \mathsf{upk}^*, m^*, \sigma_m^*, O^*)$ whose opening verifies for a pair that was never joined.

Intuitively, the only way to produce a valid opening $O$ for a signature $\sigma_m = (c, \pi)$ is to "know what the opener knows" about the ciphertext $c$: a valid decryption key $\mathsf{sk}_\Theta$ consistent with $\mathsf{pk}_\Theta$, together with a witness binding $\mathsf{sk}_\Theta$ and $c$ to the claimed identity $y$. The experiment in Fig. 10 formalizes this: the adversary is given $\mathsf{ipk}$ and oracle access to $\mathsf{Join}_\Delta$ under the honest $\mathsf{isk}$. Each oracle call returns the issued certificate $\sigma_s$ and records the pair $(\mathsf{opk}, \mathsf{upk})$ in the set $\mathcal{Q}_{\mathsf{cert}}$. To win, the adversary must output $(\mathsf{opk}^*, \mathsf{upk}^*, m^*, \sigma_m^*, O^*)$ such that $(\mathsf{opk}^*, \mathsf{upk}^*) \notin \mathcal{Q}_{\mathsf{cert}}$ but $\mathsf{Jdg}_\Delta$ accepts.

This notion rules out "phantom" openings for non-joined identities. It is orthogonal to non-frameability (which protects honest, non-signing members from being blamed) and complements trace-soundness (which rules out ambiguous openings for the same $(m, \sigma_m)$).

**Definition 13 (Opening Unforgeability).** *A group signature scheme $\Delta$ is* opening unforgeable *if for every PPT adversary $\mathcal{A}$,*

$$\Pr\left[\mathbf{Exp}_{\mathcal{A},\Delta}^{\mathsf{OpeningUnforgeability}}(\lambda) = 1\right] \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 10.

### 5.1 Secure Construction

We now present our construction and prove its security in our model. The scheme essentially mirrors the contrived construction, but with the obvious trapdoors removed, and thus closely follows the blueprint of BSZ.

```
$\mathbf{Exp}^{\mathsf{OpeningUnforgeability}}_{\mathcal{A},\Delta}(\lambda)$ :
  $\mathsf{pp}_\Delta \leftarrow_\$ \mathsf{PG}_\Delta(1^\lambda)$
  $(\mathsf{isk}, \mathsf{ipk}) \leftarrow_\$ \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$
  $\mathcal{Q}_{\mathsf{cert}} \leftarrow \emptyset$
  $(\mathsf{opk}^*, \mathsf{upk}^*, m^*, \sigma^*_m, O^*) \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{\mathsf{isk}}_{\mathsf{Join}_\Delta}}(\mathsf{ipk})$
    where oracle $\mathcal{O}^{\mathsf{isk}}_{\mathsf{Join}_\Delta}(\mathsf{opk}, \mathsf{upk})$:
      $\sigma_s \leftarrow_\$ \mathsf{Join}_\Delta(\mathsf{isk}, \mathsf{upk}, \mathsf{opk})$
      $\mathcal{Q}_{\mathsf{cert}} \leftarrow \mathcal{Q}_{\mathsf{cert}} \cup \{(\mathsf{opk}, \mathsf{upk})\}$
      return $\sigma_s$
  return $0$ if $(\mathsf{opk}^*, \mathsf{upk}^*) \in \mathcal{Q}_{\mathsf{cert}}$
  return $0$ if $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}^*, m^*, \sigma^*_m, \mathsf{upk}^*, O^*) \neq 1$
  return $1$
```

Fig. 10: Opening Unforgeability of group signature $\Delta$

We deliberately use a canonical BSZ-style instantiation. This ensures that the new properties, and not some algebraic artifacts, carry the security load. The construction demonstrates feasibility under standard assumptions.

**Languages.** We define two NP languages used in the construction. The intuition behind the scheme is identical to the contrived construction, but without the bypass. Thus, the proofs are similar.

$$L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta} := \Big\{ (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m) \;\Big|\; \exists (r, \sigma_s, w) : \begin{array}{l} \mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, f(w)), \sigma_s) = 1 \\ \wedge\ c = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, f(w); r) \Big\} \end{array}$$

$$L_2^{f,\mathsf{pp}_\Theta} := \Big\{ (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m, y) \;\Big|\; \exists \mathsf{sk}_\Theta :\ \begin{array}{l} \mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1 \\ \wedge\ \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c) = y \Big\} \end{array}$$

**Construction.** Next, we give the construction in Construction 3.

**Theorem 14.** *Let $\Delta'$ be the scheme from Construction 3. Then $\Delta'$ is perfectly correct.*

*Proof.* Follows by construction. □

```
$\mathsf{PG}_\Delta(1^\lambda)$:
  – Choose a one-way function $f$,
  – $\mathsf{pp}_\Sigma \leftarrow_\$ \mathsf{PP}_\Sigma(1^\lambda)$,
  – $\mathsf{pp}_\Theta \leftarrow_\$ \mathsf{PP}_\Theta(1^\lambda)$,
  – $\mathsf{crs}^1_\Pi \leftarrow_\$ \mathsf{PG}_\Pi(1^\lambda)$ for $L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta}$,
  – $\mathsf{crs}^2_\Pi \leftarrow_\$ \mathsf{PG}_\Pi(1^\lambda)$ for $L_2^{f,\mathsf{pp}_\Theta}$.
  – Return
    $\mathsf{pp}_\Delta = (\mathsf{crs}^1_\Pi, \mathsf{crs}^2_\Pi, \mathsf{pp}_\Sigma, \mathsf{pp}_\Theta, f)$.
$\mathsf{KgM}_\Delta(\mathsf{pp}_\Delta)$:
  – Parse
    $\mathsf{pp}_\Delta = (\mathsf{crs}^1_\Pi, \mathsf{crs}^2_\Pi, \mathsf{pp}_\Sigma, \mathsf{pp}_\Theta, f)$,
  – Sample $w \leftarrow_\$ C_\lambda$,
  – Set $y \leftarrow f(w)$,
  – Return $(\mathsf{usk}, \mathsf{upk}) = (w, y)$.
$\underline{\mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)}$:
  – Parse $\mathsf{pp}_\Delta$ as before,
  – $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_\$ \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma)$,
  – Return $(\mathsf{isk}, \mathsf{ipk}) = (\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma)$.
$\underline{\mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)}$:
  – Parse $\mathsf{pp}_\Delta$ as before.
  – $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) \leftarrow_\$ \mathsf{KG}_\Theta(\mathsf{pp}_\Theta)$,
  – Return $(\mathsf{osk}, \mathsf{opk}) = (\mathsf{sk}_\Theta, \mathsf{pk}_\Theta)$.
$\underline{\mathsf{Join}_\Delta(\mathsf{isk}, \mathsf{upk}, \mathsf{opk})}$:
  – Parse $\mathsf{sk}_\Sigma = \mathsf{isk}$, $\mathsf{upk} = y$, $\mathsf{opk} = \mathsf{pk}_\Theta$,
  – Return $\mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, y))$.
$\underline{\mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)}$:
  – Parse $\mathsf{usk} = w$, $\mathsf{upk} = y = f(w)$, $\mathsf{ipk} = \mathsf{pk}_\Sigma$, $\mathsf{opk} = \mathsf{pk}_\Theta$,
  – $(c; r) \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, f(w))$,
  – $\pi \leftarrow_\$ \mathsf{Prf}_\Pi(\mathsf{crs}^1_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m), (r, \sigma_s, w))$ for $L_1^{f,\mathsf{pp}_\Sigma,\mathsf{pp}_\Theta}$,
  – Return $\sigma_m = (c, \pi)$.
$\underline{\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m)}$:
  – Parse $(c, \pi) = \sigma_m$, $\mathsf{opk} = \mathsf{pk}_\Theta$, $\mathsf{ipk} = \mathsf{pk}_\Sigma$,
  – If $\mathsf{Vfy}_\Sigma(\mathsf{pk}'_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, y', m, c, \pi), \sigma') = 1$ return 1,
  – Return $\mathsf{Vfy}_\Pi(\mathsf{crs}^1_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m), \pi)$.
$\underline{\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m)}$:
  – If $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m) \neq 1$ return 0,
  – Parse $\mathsf{osk} = \mathsf{sk}_\Theta$, $\sigma_m = (c, \pi)$, $\mathsf{ipk} = \mathsf{pk}_\Sigma$, $\mathsf{opk} = \mathsf{pk}_\Theta$,
  – Decrypt $y \leftarrow \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c)$,
  – Generate $\pi_o \leftarrow_\$ \mathsf{Prf}_\Pi(\mathsf{crs}^2_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m, y), (\mathsf{sk}_\Theta))$ for $L_2^{f,\mathsf{pp}_\Theta}$,
  – Return $(\mathsf{upk}, O) = (y, \pi_o)$.
$\underline{\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}, O)}$:
  – If $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m) \neq 1$ return 0,
  – Return $\mathsf{Vfy}_\Pi(\mathsf{crs}^2_\Pi, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m, y), \pi_o)$.
```

Construction 3: Secure Group Signature Scheme $\Delta$

**Security.** We prove each property separately. An overview of the used assumptions is given in Table 4. For anonymity we also use traceability. We only give the intuition for the proofs of the new properties, as they are the same for the old ones.

**Theorem 15 (Trace-Soundness).** *$\Delta'$ is trace-sound.*

The proof is given in Appendix D.1.

**Theorem 16 (Non-Frameability).** *$\Delta'$ is non-frameable.*

The proof is given in Appendix D.2.

| **Property** | **ZK** | **SSE** | **SB** | **KV** | **OWF** | $\Sigma$-UNF | **DS** | $\Theta$-CPA | **Trace** |
|---|---|---|---|---|---|---|---|---|---|
| Anonymity | 2 | $q_0$ | | | | | | 1 | 1 |
| Non-frameability | 2 | 2 | | | 1 | | 1 | | |
| Traceability | 1 | 1 | | | | | | | |
| Trace-soundness | 2 | 4 | 1 | 1 | | | | | |
| Non-claimability | 1 | 1 | | | | | 1 | | |
| Cert-unforgeability | 1 | 1 | | | | 1 | | | |
| Key-unforgeability | 1 | 1 | | | $q_0$ | 1 | | | |
| Opening-unforgeability | 2 | 2 | | | | 1 | 1 | | |

Table 4: Assumptions required for each security property of $\Delta'$. Entries denote the corresponding loss; an empty cell means the assumption is not required. The parameter $q_0$ denotes the adversary's query bound for the relevant oracle (as defined in the text).

**Theorem 17 (Traceability).** $\Delta'$ *is traceable.*

The proof is given in Appendix D.3.

**Theorem 18 (Anonymity).** $\Delta'$ *is anonymous.*

The proof is given in Appendix D.4.

**Theorem 19 (Non-Claimability).** $\Delta'$ *is non-claimable.*

The proof is given in Appendix D.5.

**Theorem 20 (Certificate-unforgeability).** $\Delta'$ *is certificate-unforgeable.*

*Intuition.* Any valid group signature must embed a genuine join certificate. If the adversary outputs a forgery, either the extractor fails (breaking SSE) or it reveals a new certificate (breaking $\Sigma$ unforgeability).

The proof is given in Appendix D.6.

**Theorem 21 (Key-unforgeability).** $\Delta'$ *is key-unforgeable.*

*Intuition.* The basic reduction step is an opener needs to have a forged signature on some identity or must be able to invert a one-way function.

The proof is given in Appendix D.7.

**Theorem 22 (Opening Unforgeability).** $\Delta'$ *is opening unforgeable.*

*Intuition.* Essentially, we reduce opening unforgeability to the unforgeability of $\Sigma$ and the decryption-soundness of $\Theta$.

The proof is given in Appendix D.8.

### 5.2 On the Role of Decryption-Soundness and Key-Verifiability

Our security proofs rely on two distinct but complementary assumptions on the encryption primitive: *Decryption-Soundness* (DS) and *Key-Verifiability* (KV). While both address correctness beyond mcIND-CPA security, their scope differs fundamentally.

**Decryption-Soundness.** DS guarantees that any honestly generated ciphertext decrypts consistently under any secret key $\mathsf{sk}_\Theta$ that passes the verification test $\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1$. Formally, the experiment in Fig. 2 ensures that if $c = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m; r)$ is created honestly, then every valid $\mathsf{sk}_\Theta$ yields $\mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c) = m$. This assumption is crucial in the non-frameability and non-claimability proofs: after extracting an opening key $\mathsf{sk}^*_\Theta$ from an adversary, DS ensures that the ciphertext $c^*$ indeed decrypts to the intended $y = f(w)$ fixed during signing.

In other words, DS enforces that every honestly generated ciphertext decrypts to the same message under all verifying secret keys.

**Key-Verifiability.** KV, on the other hand, ensures functional uniqueness of secret keys. An adversary may present two distinct secret keys $\mathsf{sk}_{\Theta 0} \neq \mathsf{sk}_{\Theta 1}$ for the same $\mathsf{pk}_\Theta$, but if both pass verification $\mathsf{KVfy}_\Theta(\mathsf{sk}_{\Theta i}, \mathsf{pk}_\Theta) = 1$, then they must be *decryption-equivalent*: for all ciphertexts $c$, it holds that

$$\mathsf{Dec}_\Theta(\mathsf{sk}_{\Theta 0}, c) = \mathsf{Dec}_\Theta(\mathsf{sk}_{\Theta 1}, c).$$

Equivalently, the experiment in Fig. 1 rules out the possibility of one ciphertext $c^*$ decrypting to different values under two valid keys. This property is required in the trace-soundness proof: if two different openings were produced for the same $(m, \sigma_m)$, they would correspond to two distinct $\mathsf{sk}_\Theta$ for the same $\mathsf{pk}_\Theta$, which KV forbids.

In other words, KV enforces that all verifying secret keys are *functionally equivalent*, i.e., they decrypt any ciphertext to the same plaintext. However, KV does not imply that an encryption of some plaintext $m$ can be decrypted to some other plaintext $m^* \neq m$.

**Necessity of Both.** Neither assumption subsumes the other:

- DS alone does not prevent *key ambiguity* (two valid keys disagreeing on decryption), since it only speaks about honestly generated ciphertexts.
- KV alone does not guarantee that honest ciphertexts are decrypted correctly; it only ensures that *if* two keys are valid, they are functionally equivalent.

Thus both are simultaneously required: DS underpins the integrity of honest encryptions in non-frameability and non-claimability, while KV ensures uniqueness of openings in trace-soundness.

## 6 Separations of the Definitions

We now show that each of the seven security definitions is independent.

In other words, for each property there exists a contrived scheme which achieves all but that one property.

**Theorem 23 (Independence of Anonymity).** *There exists a group signature scheme $\Delta^{\neg\mathsf{ANON}}$ that is non-frameable, traceable, trace-sound, non-claimable, certificate-unforgeable, key-unforgeable, and opening-unforgeable, but not anonymous.*

The proof is given in Appendix E.3.

**Intuition.** We leak the signer's public identity by appending $\mathsf{upk}$ *in clear* to the output, while delegating *all* validity checks to the baseline scheme $\Delta$. Concretely, we redefine the message space as $m' \leftarrow (m, \mathsf{upk})$.

**Theorem 24 (Independence of Trace-Soundness).** *There exists a group signature scheme $\Delta^{\neg\mathsf{TS}}$ that satisfies anonymity, non-frameability, traceability, non-claimability, certificate-unforgeability, key-unforgeability, and opening-unforgeability, but is not trace-sound.*

The proof is given in Appendix E.2.

**Intuition.** We provide a counter example. We equip the *issuer* and the *opener* public keys with a single public marker—a special raw signature value $\sigma^\star$, which we assume to be outside the signature space, e.g., $0$. If *both* keys carry the *same* non-$\perp$ marker, then $\mathsf{Vfy}'_\Delta$ unconditionally accepts any transcript whose signature component equals $\sigma^\star$ (for *any* message), while $\mathsf{Opn}'_\Delta$ returns $\perp$ and $\mathsf{Jdg}'_\Delta$ accepts *any* claimed identity in the two-element set $\{0, 1\}$. (We assume $0$ and $1$ are not valid member public keys.) When at least one authority is honest, the marker is $\perp$ and the wrapper collapses to the baseline.

Formally, we keep the syntactic context binding of the public keys to the message.

**Theorem 25 (Independence of Traceability).** *There exists a group signature scheme $\Delta^{\neg\mathsf{TRC}}$ that satisfies anonymity, non-frameability, trace-soundness, non-claimability, certificate-unforgeability, key-unforgeability, and opening-unforgeability, but is not traceable.*

The proof is given in Appendix E.3.

**Intuition.** We provide a concrete counterexample. We augment only the *issuer* public key with a single flag bit $b_{\mathsf{trc}} \in \{0, 1\}$. If the issuer is honest, it publishes $b_{\mathsf{trc}} = 0$. A corrupt issuer may set $b_{\mathsf{trc}} = 1$. Verification is unchanged. When $b_{\mathsf{trc}} = 1$ and a signature verifies, the opener interfaces are hardwired to refuse: $\mathsf{Opn}'_\Delta$ returns $\perp$ and $\mathsf{Jdg}'_\Delta$ returns $0$. Hence, valid but unopenable signatures exist.

**Theorem 26 (Independence of Non-Frameability).** *There exists a group signature scheme $\Delta^{\neg\mathsf{NF}}$ that satisfies anonymity, traceability, trace-soundness, non-claimability, certificate-unforgeability, key-unforgeability, and opening-unforgeability, but is not non-frameable.*

**Intuition.** We provide a counterexample. We force every honest signer to append a trailing $0$ to its message, so all honest transcripts are of the form $(m, 0)$. The wrapper then accepts *both* $(m, 0)$ and $(m, 1)$ in verification. If the issuer and opener both publish the same non-$\perp$ anchor identity $y^\star$ in their public keys, then $\mathsf{Jdg}'_\Delta$ reroutes *all* signatures on $(m, 1)$ to $y^\star$, ignoring the actual witness. Thus, colluding authorities can frame an honest member by pointing the anchor at its identity, while honest authorities ($\perp$ anchors) or honest transcripts $(m, 0)$ collapse to the baseline behavior.

The proof is given in Appendix E.4.

**Theorem 27 (Independence of Non-Claimability).** *There exists a group signature scheme $\Delta^{\neg\mathsf{NC}}$ that satisfies anonymity, non-frameability, traceability, trace-soundness, certificate-unforgeability, key-unforgeability, and opening-unforgeability, but is not non-claimable. Moreover, breaking NC in our wrapper necessarily requires collusion of both issuer and opener.*

**Intuition.** A corrupt issuer and opener jointly publish the same non-$\perp$ randomness $r$, which fixes a target key $(\mathsf{usk}^\dagger, \mathsf{upk}^\dagger) \leftarrow \mathsf{KgM}_\Delta(r)$ that never joined. For any honestly logged transcript $(m, \sigma')$ of the single honest member, the modified $\mathsf{Jdg}'_\Delta$ accepts a *claim* to $\mathsf{upk}^\dagger$ *iff* the supplied witness embeds a baseline-valid opening for the true signer (checked on the baseline), and both authorities committed to the same $r$. Thus, colluding authorities can *relabel* any honest transcript to the adversarial key $\mathsf{upk}^\dagger$, violating NC. Outside this gated branch the wrapper delegates to the baseline, so the remaining seven properties persist.

The proof is given in Appendix E.5.

**Theorem 28 (Independence of Certificate Unforgeability).** *There exists a group signature scheme $\Delta^{\neg\mathsf{CUnf}}$ that satisfies anonymity, non-frameability, traceability, trace-soundness, non-claimability, key-unforgeability, and opening-unforgeability, but is not certificate-unforgeable.*

**Intuition.** We equip the opener's public key with a switch bit $b_{\mathsf{cunf}}$. If $b_{\mathsf{cunf}} = 0$ the wrapper reduces to the baseline. If $b_{\mathsf{cunf}} = 1$, honest signers ignore their input and always sign $h = f(\mathsf{usk})$, where $\mathsf{usk}$ is their membership secret key and $f$ is a one-way function. Thus, the adversary can use a signing-oracle transcript to derive a fresh, valid "certificate transcript" on $h$ which was never produced by the honest join protocol, breaking CUnf. Since the modification affects only the message space of signatures and disables openings in the gadget branch, all other properties continue to hold by straight-line reduction to the baseline scheme.

The proof is given in Appendix E.6.

**Theorem 29 (Independence of Key Unforgeability).** *There exists a group signature scheme $\Delta^{\neg\mathsf{KUnf}}$ that satisfies anonymity, non-frameability, traceability, trace-soundness, certificate-unforgeability, non-claimability, and opening-unforgeability, but is not key-unforgeable.*

**Intuition.** We equip the opener's public key with a switch bit $b_{\mathsf{kunf}}$. If $b_{\mathsf{kunf}} = 0$ the wrapper reduces to the baseline. If $b_{\mathsf{kunf}} = 1$, every honest signer ignores the external message $m$ and instead signs $h = f(\sigma_s)$, where $\sigma_s$ is its join-certificate and $f$ is a one-way function. The opening interfaces are hardwired to reject, i.e., $\mathsf{Opn}'_\Delta$ returns $\perp$ and $\mathsf{Jdg}'_\Delta$ returns $0$. A corrupt opener, given any certificate $\sigma_s$ issued by the honest authority, can compute $h = f(\sigma_s)$ and mint fresh valid signatures on $h$ without access to any member secret key, thereby breaking KUnf. All other properties reduce to the baseline and remain intact.

The proof is given in Appendix E.7.

**Theorem 30 (Independence of Opening-Unforgeability).** *There exists a group signature scheme $\Delta^{\neg\mathsf{OpU}}$ that satisfies anonymity, non-frameability, traceability, trace-soundness, non-claimability, key-unforgeability, and certificate-unforgeability, but is not opening-unforgeable.*

**Intuition.** We provide a counterexample. The opener's public key is extended with an auxiliary string $r$. If the opener is honest it sets $r \leftarrow \perp$ and the wrapper reduces to the baseline scheme. If the opener is corrupt (as in the relevant OpU game), it may publish any $r \neq \perp$, which in turn fixes a deterministic member key pair $(\mathsf{usk}^{\dagger}, \mathsf{upk}^{\dagger}) \leftarrow \mathsf{KgM}_{\Delta}(r)$.

Honest users always sign messages in the form $(m, 0)$. Corrupted (but joined) users may also produce signatures on $(m, 1)$. Verification accepts either flag. The wrapper, however, remaps openings for flag-1 signatures: whenever a transcript on $(m, 1)$ verifies, the $\mathsf{OpenVerify}$ algorithm ignores the supplied opening witness and instead accepts the claim that the signer was exactly the member $\mathsf{upk}^{\dagger}$ derived from $r$.

This allows an adversary to take any valid signature on $(m, 1)$ from a joined member and turn it into an accepted opening for the never-joined $\mathsf{upk}^{\dagger}$. Hence OpU is violated. For all other properties the wrapper is inert (on honest transcripts with flag 0) or reduces directly to the base scheme, so anonymity, non-frameability, traceability, trace-soundness, non-claimability, key-unforgeability, and certificate-unforgeability remain intact.

The proof is given in Appendix E.8.

**Acknowledgements**. This work was in part supported by the European Union's Horizon Europe project LICORICE (grant agreement no. 101168311) and the CHIST-ERA project REMINDER through the Austrian Science Fund (FWF) project number I 6650-N. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the funding agencies. Neither European Union nor the granting authorities can be held responsible for them.

## References

1. Alamélou, Q., Blazy, O., Cauchie, S., Gaborit, P.: A code-based group signature scheme. Des. Codes Cryptogr. 82(1-2), 469–493 (2017), https://doi.org/10.1007/s10623-016-0276-6
2. Backes, M., Hanzlik, L., Schneider-Bensch, J.: Membership privacy for fully dynamic group signatures. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019. pp. 2181–2198. ACM (2019), https://doi.org/10.1145/3319535.3354257
3. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 259–274. Springer (2000), https://doi.org/10.1007/3-540-45539-6_18

4. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2656, pp. 614–629. Springer (2003), https://doi.org/10.1007/3-540-39200-9_38
5. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: The case of dynamic groups. In: Menezes, A. (ed.) Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3376, pp. 136–153. Springer (2005), https://doi.org/10.1007/978-3-540-30574-3_11
6. Benhamouda, F., Camenisch, J., Krenn, S., Lyubashevsky, V., Neven, G.: Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 551–572. Springer (2014), https://doi.org/10.1007/978-3-662-45611-8_29
7. Bichsel, P., Camenisch, J., Neven, G., Smart, N.P., Warinschi, B.: Get shorty via group signatures without encryption. In: Garay, J.A., Prisco, R.D. (eds.) Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6280, pp. 381–398. Springer (2010), https://doi.org/10.1007/978-3-642-15317-4_24
8. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6597, pp. 253–273. Springer (2011), https://doi.org/10.1007/978-3-642-19571-6_16
9. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004. pp. 168–177. ACM (2004), https://doi.org/10.1145/1030083.1030106
10. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. In: Manulis, M., Sadeghi, A., Schneider, S.A. (eds.) Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9696, pp. 117–136. Springer (2016), https://doi.org/10.1007/978-3-319-39555-5_7
11. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. J. Cryptol. 33(4), 1822–1870 (2020), https://doi.org/10.1007/s00145-020-09357-w
12. Camenisch, J., Lehmann, A., Neven, G., Samelin, K.: Uc-secure non-interactive public-key encryption. In: 30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017. pp. 217–233. IEEE Computer Society (2017), https://doi.org/10.1109/CSF.2017.14
13. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings. Lecture Notes in Computer Science, vol. 547, pp. 257–265. Springer (1991), https://doi.org/10.1007/3-540-46416-6_22

14. Delerablée, C., Pointcheval, D.: Dynamic fully anonymous short group signatures. In: Nguyen, P.Q. (ed.) Progressin Cryptology - VIETCRYPT 2006, First International Conference on Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4341, pp. 193–210. Springer (2006), https://doi.org/10.1007/11958239_13
15. Derler, D., Samelin, K., Slamanig, D.: Bringing order to chaos: The case of collision-resistant chameleon-hashes. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12110, pp. 462–492. Springer (2020), https://doi.org/10.1007/978-3-030-45374-9_16
16. Derler, D., Slamanig, D.: Highly-efficient fully-anonymous dynamic group signatures. In: Kim, J., Ahn, G., Kim, S., Kim, Y., López, J., Kim, T. (eds.) Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018. pp. 551–565. ACM (2018), https://doi.org/10.1145/3196494.3196507
17. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the fiat-shamir transform. In: Galbraith, S.D., Nandi, M. (eds.) Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7668, pp. 60–79. Springer (2012), https://doi.org/10.1007/978-3-642-34931-7_5
18. Furukawa, J., Imai, H.: An efficient group signature scheme from bilinear maps. In: Boyd, C., Nieto, J.M.G. (eds.) Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3574, pp. 455–467. Springer (2005), https://doi.org/10.1007/11506157_38
19. Groth, J.: Fully anonymous group signatures without random oracles. In: Kurosawa, K. (ed.) Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4833, pp. 164–180. Springer (2007), https://doi.org/10.1007/978-3-540-76900-2_10
20. Hofheinz, D., Rao, V., Wichs, D.: Standard security does not imply indistinguishability under selective opening. In: Hirt, M., Smith, A.D. (eds.) Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9986, pp. 121–145 (2016), https://doi.org/10.1007/978-3-662-53644-5_5
21. Krenn, S., Samelin, K., Striecks, C.: Practical group-signatures with privacy-friendly openings. In: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES 2019, Canterbury, UK, August 26-29, 2019. pp. 10:1–10:10. ACM (2019), https://doi.org/10.1145/3339252.3339256
22. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2442, pp. 111–126. Springer (2002), https://doi.org/10.1007/3-540-45708-9_8
23. Sakai, Y., Schuldt, J.C.N., Emura, K., Hanaoka, G., Ohta, K.: On the security of dynamic group signatures: Preventing signature hijacking. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Pro-

ceedings. Lecture Notes in Computer Science, vol. 7293, pp. 715–732. Springer (2012), https://doi.org/10.1007/978-3-642-30057-8_42

## A Primitives

**One-Way Functions.** One-way functions formalize the intuitive notion of a function $f$ that is easy to compute but hard to invert.

Let $\mathcal{F} = \{f_\lambda : C_\lambda \to D_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of efficiently computable functions, where $C_\lambda$ and $D_\lambda$ are some domains (e.g., $\mathbb{Z}_q$ or $\{0,1\}^n$). $C_\lambda$ must be efficiently samplable.

**Definition 14 (One-Way Function).** *We say that $\mathcal{F}$ is a one-way function family if for all PPT adversaries $\mathcal{A}$, the probability*

$$\Pr\left[\begin{matrix} x \leftarrow_{\$} C_\lambda;\ y \leftarrow f_\lambda(x); \\ x' \leftarrow_{\$} \mathcal{A}(1^\lambda, y) : f_\lambda(x') = y \end{matrix}\right] \leq \nu(\lambda)$$

*is negligible.*

**Digital Signature Schemes.** A digital signature scheme allows a party to sign messages such that the authenticity of the signature can be publicly verified.

**Framework.** We now present the formal framework for digital signature schemes.

**Definition 15 (Digital Signature Scheme).** *A digital signature scheme $\Sigma$ is a tuple of four PPT algorithms $(\mathsf{PP}_\Sigma, \mathsf{KG}_\Sigma, \mathsf{Sgn}_\Sigma, \mathsf{Vfy}_\Sigma)$, such that:*

$\mathsf{PP}_\Sigma$**.** *On input $1^\lambda$, output public parameters $\mathsf{pp}_\Sigma$:*

$$\mathsf{pp}_\Sigma \leftarrow_{\$} \mathsf{PP}_\Sigma(1^\lambda).$$

*We assume that $\mathsf{pp}_\Sigma$ is implicit input to all other algorithms.*

$\mathsf{KG}_\Sigma$**.** *On input $\mathsf{pp}_\Sigma$, output signing and verification keys $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma)$:*

$$(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_{\$} \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma).$$

$\mathsf{Sgn}_\Sigma$**.** (Possibly randomized.) *On input $(\mathsf{sk}_\Sigma, m)$ output $\sigma$:*

$$\sigma \leftarrow_{\$} \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, m).$$

$\mathsf{Vfy}_\Sigma$**.** (Deterministic.) *On input $(\mathsf{pk}_\Sigma, m, \sigma)$ output $d \in \{0,1\}$:*

$$d \leftarrow \mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, m, \sigma).$$

**Definition 16 (Correctness).** *A digital signature scheme is correct if for all $\lambda \in \mathbb{N}$, for all $\mathsf{pp}_\Sigma \leftarrow_{\$} \mathsf{PP}_\Sigma(1^\lambda)$, for all $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_{\$} \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma)$, and all $m \in \mathcal{MS}$,*

$$\mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, m, \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, m)) = 1.$$

```
$\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\lambda)$ :
    $\mathsf{pp}_\Sigma \leftarrow_\$ \mathsf{PP}_\Sigma(1^\lambda)$
    $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_\$ \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma)$
    $\mathcal{Q} \leftarrow \emptyset$
    $(m^*, \sigma^*) \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{\mathsf{sk}_\Sigma}_{\mathsf{Sign}}}(\mathsf{pk}_\Sigma)$
        where $\mathcal{O}^{\mathsf{sk}_\Sigma}_{\mathsf{Sign}}(m)$:
            $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
            return $\sigma \leftarrow_\$ \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, m)$
    return $1$ if $\mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, m^*, \sigma^*) = 1 \wedge m^* \notin \mathcal{Q}$
    return $0$
```

Fig. 11: Existential Unforgeability under Chosen Message Attacks of signature scheme $\Sigma$

**Security.** We use unforgeability under chosen-message attacks.

**Unforgeability.** Unforgeability requires that an adversary cannot come up with a signature on a never signed message, even with adaptive oracle access to a signing oracle.

**Definition 17 (Unforgeability under Chosen Message Attacks).** *A digital signature scheme $\Sigma$ is existentially unforgeable under chosen message attacks (EUF-CMA) if for every PPT adversary $\mathcal{A}$, there exists a negligible function $\nu$ such that:*

$$\Pr\Big[\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\lambda) = 1\Big] \leq \nu(\lambda).$$

Where the corresponding experiment is given in Figure 11.

**Decryption-Sound Key-Verifiable Public Key Encryption Schemes. Framework.** We now present the formal framework.

**Definition 18 (Key-Verifiable and Decryption-Sound Encryption Schemes).** (Intuition: honest encryptions decrypt consistently and keys are checkable.) *A decryption-sound encryption scheme $\Theta$ is a tuple of PPT algorithms* $(\mathsf{PP}_\Theta, \mathsf{KG}_\Theta, \mathsf{Enc}_\Theta, \mathsf{Dec}_\Theta, \mathsf{KVfy}_\Theta)$*, such that:*

$\mathsf{PP}_\Theta$**.** *On input $1^\lambda$, output public parameters $\mathsf{pp}_\Theta$:*

$$\mathsf{pp}_\Theta \leftarrow_\$ \mathsf{PP}_\Theta(1^\lambda).$$

$\mathsf{KG}_\Theta$**.** *On input $\mathsf{pp}_\Theta$, output a decryption key $\mathsf{sk}_\Theta$ and encryption key $\mathsf{pk}_\Theta$:*

$$(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) \leftarrow_\$ \mathsf{KG}_\Theta(\mathsf{pp}_\Theta).$$

$\mathsf{Enc}_\Theta$**.** *(Randomized) encryption on input $\mathsf{pk}_\Theta$ and $m$ samples internal randomness $r$ and outputs a ciphertext $c$:*

$$c \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m).$$

$\mathsf{Dec}_\Theta$. *Deterministic decryption on input* $\mathsf{sk}_\Theta$ *and* $c$ *outputs* $m$*:*

$$m \leftarrow \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c).$$

$\mathsf{KVfy}_\Theta$. *Deterministic key verification on input* $\mathsf{sk}_\Theta, \mathsf{pk}_\Theta$ *outputs* $d \in \{0,1\}$*:*

$$d \leftarrow \mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta).$$

**Definition 19 (Correctness).** (Intuition: decrypting an honest ciphertext recovers the message; keys verify.) *$\Theta$ is correct if for all $\lambda \in \mathbb{N}$, all $\mathsf{pp}_\Theta \leftarrow_\$ \mathsf{PP}_\Theta(1^\lambda)$, all $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) \leftarrow_\$ \mathsf{KG}_\Theta(\mathsf{pp}_\Theta)$, and all $m \in \mathcal{M}$,*

$$m = \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m)) \quad \textit{and} \quad \mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1.$$

**Security.** CPA for $\Theta$ requires that an adversary cannot distinguish encryptions of two chosen messages. We use a fully adaptive version. A standard hybrid argument shows polynomial equivalence to the single-challenge form [3].

**CPA-Security.** A public-key encryption scheme $\Theta$ is CPA-secure if no efficient adversary can distinguish left/right encryptions.

```
$\mathbf{Exp}^{\mathsf{mcINDCPA}}_{\mathcal{A},\Theta}(\lambda)$ :
  $\mathsf{pp}_\Theta \leftarrow_\$ \mathsf{PP}_\Theta(1^\lambda)$
  $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) \leftarrow_\$ \mathsf{KG}_\Theta(\mathsf{pp}_\Theta)$
  $b \leftarrow_\$ \{0,1\}$
  $b^* \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{b,\mathsf{pk}_\Theta}_{\mathsf{Enc}'_\Theta}}(\mathsf{pk}_\Theta)$
    where $\mathcal{O}^{b,\mathsf{pk}_\Theta}_{\mathsf{Enc}'_\Theta}(m_0, m_1)$:
      $c_0 \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m_0)$
      $c_1 \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m_1)$
      return $\perp$ if $c_0 = \perp \vee c_1 = \perp$
      return $c_b$
  return 1 if $b^* = b$
  return 0
```

Fig. 12: Chosen-Plaintext Attack (CPA) security of PKE $\Theta$

**Definition 20 (Chosen-Plaintext Attacks).** *$\Theta$ is CPA-secure if for every PPT adversary $\mathcal{A}$ there exists a negligible $\nu$ such that*

$$\left|\Pr\left[\mathbf{Exp}^{\mathsf{mcINDCPA}}_{\mathcal{A},\Theta}(\lambda) = 1\right] - 1/2\right| \leq \nu(\lambda).$$

$\underline{\mathsf{PP}_\Theta(1^\lambda)}$ : Run $(\mathbb{G}, q, g) \leftarrow \mathsf{GGen}(1^\lambda)$ with $\mathbb{G}$ cyclic of prime order $q$ and generator $g$. Return $\mathsf{pp}_\Theta \leftarrow (\mathbb{G}, q, g)$.

$\underline{\mathsf{KG}_\Theta(\mathsf{pp}_\Theta)}$ : Sample $x \leftarrow_\$ \mathbb{Z}_q$. Let $\mathsf{sk}_\Theta \leftarrow x$ and $\mathsf{pk}_\Theta \leftarrow g^x \in \mathbb{G}$. Return $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta)$.

$\underline{\mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m)}$ : Sample $r \leftarrow_\$ \mathbb{Z}_q$ and output

$$c \leftarrow (c_1, c_2) \leftarrow (g^r,\ m \cdot \mathsf{pk}_\Theta^r) \in \mathbb{G}^2.$$

$\underline{\mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c)}$ : Parse $c = (c_1, c_2)$ and return $m \leftarrow c_2 / c_1^{\mathsf{sk}_\Theta}$.

$\underline{\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta)}$ : Return 1 if $\mathsf{pk}_\Theta = g^{\mathsf{sk}_\Theta}$, else 0.

Construction 4: ElGamal Encryption over Prime-Order Groups

Where the corresponding experiment is given in Figure 12.

**Instantiation.** We now present a suitable instantiation - ElGamal. Here $\mathcal{M} = \mathbb{G}$ and the randomness space is $\mathbb{Z}_q$. See Construction 4.

**Theorem 31.** *Let $\Theta$ be as in Construction 4. Then $\Theta$ is perfectly correct.*

*Proof.* Immediate from the equations. □

**Theorem 32.** *Let $\Theta$ be as in Construction 4. Then $\Theta$ is mcINDCPA-secure.*

*Proof.* Standard under DDH. □

**Non-Interactive Proof Systems.** Let $L$ be an NP-language with associated witness relation $R$. A non-interactive proof system allows proving membership $x \in L$ without interaction.

**Parameterized NP-Languages.** In many applications, the relation depends on externally chosen parameters. We model this as a family $L^{\mathcal{F}}$, indexed by a parameter $f \in \mathcal{F}$:

$$L^{\mathcal{F}} := \big\{((f, x), w) \bigm| R_f(x, w) = 1\big\},$$

where $(f, x)$ is the statement. The parameter is fixed at setup (e.g., via the CRS). All NIZK notions are understood relative to the induced language. For readability we omit the parameter when unambiguous; constructions are families parameterized by $L$.

**Framework.** We now state the interface.

**Definition 21 (Non-Interactive Proof System).** *A non-interactive proof system $\Pi$ for language $L$ consists of PPT algorithms $\{\mathsf{PG}_\Pi, \mathsf{Prf}_\Pi, \mathsf{Vfy}_\Pi\}$:*

$\mathsf{PG}_\Pi$**.** *On input $1^\lambda$, output $\mathsf{crs}_\Pi$:*

$$\mathsf{crs}_\Pi \leftarrow_\$ \mathsf{PG}_\Pi(1^\lambda).$$

$\mathsf{Prf}_\Pi$**.** *On input $(\mathsf{crs}_\Pi, x, w)$ output a proof $\pi$:*

$$\pi \leftarrow_\$ \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, x, w).$$

$\mathsf{Vfy}_\Pi$. *On input* $(\mathsf{crs}_\Pi, x, \pi)$ *output* $d \in \{0,1\}$*:*

$$d \leftarrow \mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x, \pi).$$

**Definition 22 (Correctness).** *For all* $\lambda \in \mathbb{N}$*, all* $\mathsf{crs}_\Pi \leftarrow_\$ \mathsf{PG}_\Pi(1^\lambda)$*, all* $x \in L$*, all* $w$ *with* $R(x, w) = 1$*, and all* $\pi \leftarrow_\$ \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, x, w)$*, we have* $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x, \pi) = 1$.

We additionally require zero-knowledge and simulation-sound extractability.

**Security.** We present the guarantees we require.

**Zero-Knowledge.** Informally, the verifier learns nothing beyond validity.

```
$\mathbf{Exp}^{\mathsf{Zero\text{-}Knowledge}}_{\mathcal{A},\Pi,\mathsf{SIM}}(\lambda)$ :
  $(\mathsf{crs}_\Pi, \tau) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$
  $b \leftarrow_\$ \{0,1\}$
  $b^* \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{\tau}_{P_b}}(\mathsf{crs}_\Pi)$
    where $\mathcal{O}^{\tau}_{P_0}(x, w)$:
      return $\perp$ if $R(x, w) \neq 1$
      return $\mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, x, w)$
    where $\mathcal{O}^{\tau}_{P_1}(x, w)$:
      return $\perp$ if $R(x, w) \neq 1$
      return $\mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, x)$
  return 1 if $b^* = b$; else 0
```

Fig. 13: Zero-Knowledge of NIZK proof system $\Pi$

**Definition 23 (Zero-Knowledge).** *A non-interactive proof system* $\Pi$ *for* $L$ *is zero-knowledge if there exists a PPT simulator* $\mathsf{SIM} = (\mathsf{SIM}_1, \mathsf{SIM}_2)$ *such that for every PPT* $\mathcal{A}$ *there are negligible functions* $\nu_1, \nu_2$ *with*

$$\left|\Pr[\mathsf{crs}_\Pi \leftarrow_\$ \mathsf{PG}_\Pi(1^\lambda) : \mathcal{A}(\mathsf{crs}_\Pi) = 1] - \Pr[(\mathsf{crs}_\Pi, \tau) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda) : \mathcal{A}(\mathsf{crs}_\Pi) = 1]\right| \leq \nu_1(\lambda),$$

*and*

$$\left|\Pr\left[\mathbf{Exp}^{\mathsf{Zero\text{-}Knowledge}}_{\mathcal{A},\Pi,\mathsf{SIM}}(\lambda) = 1\right] - 1/2\right| \leq \nu_2(\lambda).$$

Note, we do not require composible zero-knowledge as defined by Groth [19].

**Simulation-Sound Extractability.** Even after seeing simulated proofs for adaptively chosen statements, any accepting proof on a fresh statement admits extraction of a valid witness.

```
$\mathbf{Exp}^{\mathsf{SimSoundExt}}_{\mathcal{A},\Pi,\mathcal{E}}(\lambda)$
  $(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_\$ \mathcal{E}_1(1^\lambda)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(x^*, \pi^*) \leftarrow_\$ \mathcal{A}^{\mathcal{O}^\tau_{\mathsf{SIM}}(\cdot)}(\mathsf{crs}_\Pi)$
    where $\mathcal{O}^\tau_{\mathsf{SIM}}(x)$:
      $\pi \leftarrow_\$ \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, x)$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(x, \pi)\}$
      return $\pi$
  $w^* \leftarrow_\$ \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, x^*, \pi^*)$
  return 1 if $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x^*, \pi^*) = 1 \ \wedge\ R(x^*, w^*) = 0 \ \wedge\ (x^*, \pi^*) \notin \mathcal{Q}$
  return 0
```

Fig. 14: Simulation-Sound Extractability of NIZK proof system $\Pi$

**Definition 24 (Simulation-Sound Extractability).** *A NIZK $\Pi$ for $L$ is simulation-sound extractable if there exists a PPT extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ such that for every PPT $\mathcal{A}$,*

$$\left|\Pr[(\mathsf{crs}_\Pi, \tau) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda) : \mathcal{A}(\mathsf{crs}_\Pi) = 1] - \Pr[(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_\$ \mathcal{E}_1(1^\lambda) : \mathcal{A}(\mathsf{crs}_\Pi) = 1]\right| = 0,$$

*and there exists a negligible function $\nu$ with*

$$\Pr\left[\mathbf{Exp}^{\mathsf{SimSoundExt}}_{\mathcal{A},\Pi,\mathcal{E}}(\lambda) = 1\right] \leq \nu(\lambda),$$

*where $\mathsf{SIM}$ is as in Def. 23.*

# B Proofs for the Encryption Scheme

## B.1 Proof of Theorem 1 (Key-Verifiability)

*Proof.* We have $\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1$ iff $\mathsf{pk}_\Theta = g^{\mathsf{sk}_\Theta}$. In a prime-order group, the map $x \mapsto g^x$ is injective modulo $q$, hence there cannot exist two different $\mathsf{sk}_\Theta$ that both verify for the same $\mathsf{pk}_\Theta$. Consequently, any two verifying keys for a fixed $\mathsf{pk}_\Theta$ must be identical and therefore decrypt every ciphertext to the same value. The experiment in Figure 1 thus returns 1 with probability 0. □

## B.2 Proof of Theorem 2 (Decryption-Soundness)

*Proof.* If $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1$, then $\mathsf{pk}^*_\Theta = g^{\mathsf{sk}^*_\Theta}$. For $c = (g^{r^*}, m^* \cdot (\mathsf{pk}^*_\Theta)^{r^*})$,

$$\mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c) = \frac{m^* \cdot (g^{\mathsf{sk}^*_\Theta})^{r^*}}{(g^{r^*})^{\mathsf{sk}^*_\Theta}} = m^*.$$

Thus Figure 2 returns 1 with probability 0. □

### B.3 Proof of Theorem 3 (Key-Verifiability does not imply Decryption-Soundness)

**Starting point.** Let $\Theta_0 = (\mathsf{PP}_\Theta, \mathsf{KG}_\Theta, \mathsf{Enc}_\Theta, \mathsf{Dec}_\Theta, \mathsf{KVfy}_\Theta)$ be any correct PKE with message space $\mathcal{M}$. Fix some $m' \in \mathcal{M}$.

**Construction of $\Theta'$.** Add a tag $t \in \{0,1\}^*$ to keys and force decryption to $t$ if $t \neq \bot$.

- **Parameters.** As in $\Theta_0$.
- **KeyGen.** Honest generation fixes $b = \bot$:

$$\mathsf{pk}'_\Theta = (\mathsf{pk}_\Theta, \bot), \qquad \mathsf{sk}'_\Theta = (\mathsf{sk}_\Theta, \bot),$$

  where $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) \leftarrow_\$ \mathsf{KG}_\Theta(\mathsf{pp}_\Theta)$.
- **Key verification.**

$$\mathsf{KVfy}'_\Theta\big((\mathsf{sk}_\Theta, t), (\mathsf{pk}_\Theta, t')\big) = 1 \iff \mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1 \wedge t = t'.$$

- **Encryption.** $\mathsf{Enc}'_\Theta\big((\mathsf{pk}_\Theta, t), m\big) \leftarrow \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m)$.
- **Decryption (twist).**

$$\mathsf{Dec}'_\Theta\big((\mathsf{sk}_\Theta, t), c\big) \leftarrow \begin{cases} \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c), & t = \bot, \\ t & t \neq \bot. \end{cases}$$

*Remark 2.* If honest keys always have $t = \bot$ then $\Theta'$ inherits correctness from $\Theta_0$. In the Decryption Soundness experiment (Fig. 2), however, the adversary is free to choose keys; picking $t \neq \bot$ breaks decryption soundness by design.

*Proof.* Consider $\Theta'$ as given above. Let the adversary output

$$(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta, m^*) = ((\mathsf{sk}_\Theta, 1), (\mathsf{pk}_\Theta, 1), m^*)$$

for any base pair with $\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1$ and any $m^* \neq m'$. Then

$$\mathsf{KVfy}'_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1 \text{ and } c \leftarrow_\$ \mathsf{Enc}'_\Theta(\mathsf{pk}^*_\Theta, m^*) = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m^*),$$

but $\mathsf{Dec}'_\Theta(\mathsf{sk}^*_\Theta, c) = m' \neq m^*$. Hence $\mathbf{Exp}^{\mathsf{DecSound}}_{\mathcal{A},\Theta'}(\lambda) = 1$ with probability 1, while key verifiability still holds by the tag check. □

### B.4 Proof of Theorem 4 (Decryption-Soundness does not imply Key-Verifiability)

We now separate decryption soundness from key verifiability by giving a scheme that is decryption-sound but *not* key-verifiable.

**Starting point.** Let $\Theta_0 = (\mathsf{PP}_\Theta, \mathsf{KG}_\Theta, \mathsf{Enc}_\Theta, \mathsf{Dec}_\Theta, \mathsf{KVfy}_\Theta)$ be any correct PKE with message space $\mathcal{M} \subseteq \{0,1\}^\ell$ (for some $\ell \geq 1$). Fix a publicly known involution $\tau : \mathcal{M} \to \mathcal{M}$ with $\tau \neq \mathrm{id}$ (e.g., flip the last bit).

**Construction of $\Theta''$.** We append a one-bit *flag* $f \in \{0,1\}$ to ciphertexts and a one-bit *tag* $b \in \{0,1\}$ to secret keys. Encryption always sets $f = 0$; decryption *may* react to $f = 1$ depending on $b$.

- **Parameters.** As in $\Theta_0$.
- **KeyGen.** Sample $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) \leftarrow_\$ \mathsf{KG}_\Theta(\mathsf{pp}_\Theta)$ and $b \leftarrow_\$ \{0,1\}$. Output

$$\mathsf{sk}''_\Theta = (\mathsf{sk}_\Theta, b), \qquad \mathsf{pk}''_\Theta = \mathsf{pk}_\Theta.$$

- **Key verification.**

$$\mathsf{KVfy}''_\Theta\big((\mathsf{sk}_\Theta, b), \mathsf{pk}_\Theta\big) \leftarrow \mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta).$$

(Note: $b$ is ignored by verification.)

- **Encryption.** Always set the flag $f = 0$ and encrypt under $\Theta_0$:

$$\mathsf{Enc}''_\Theta(\mathsf{pk}_\Theta, m) \leftarrow (f, c_0) \ \text{ with } \ f \leftarrow 0,\ c_0 \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m).$$

- **Decryption.** Parse $c = (f, c_0)$ and return

$$\mathsf{Dec}''_\Theta\big((\mathsf{sk}_\Theta, b), (f, c_0)\big) \leftarrow \begin{cases} \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c_0), & f = 0, \\ \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c_0), & f = 1 \text{ and } b = 0, \\ \tau\big(\mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c_0)\big), & f = 1 \text{ and } b = 1. \end{cases}$$

*Proof. Decryption soundness.* In $\mathbf{Exp}^{\mathsf{DecSound}}$ (Fig. 2), the challenger computes $c \leftarrow_\$ \mathsf{Enc}''_\Theta(\mathsf{pk}^*_\Theta, m^*; r^*)$. By construction $\mathsf{Enc}''_\Theta$ *always* sets $f = 0$, hence

$$\mathsf{Dec}''_\Theta\big((\mathsf{sk}^*_\Theta, b^*), c\big) = \mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c_0) = m^*,$$

for any $b^* \in \{0,1\}$, whenever $\mathsf{KVfy}''_\Theta((\mathsf{sk}^*_\Theta, b^*), \mathsf{pk}^*_\Theta) = 1$. Thus the experiment outputs $1$ with probability $0$ and $\Theta''$ is decryption-sound.

*Failure of Key verifiability.* In $\mathbf{Exp}^{\mathsf{KeyVerifiability}}$ (Fig. 1), the adversary proceeds as follows: pick any $(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta)$ with $\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta, \mathsf{pk}_\Theta) = 1$, choose any $m \in \mathcal{M}$ and an arbitrary $c_0 \leftarrow_\$ \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, m)$, and set

$$\mathsf{sk}_\Theta{}^*_0 \leftarrow (\mathsf{sk}_\Theta, 0), \quad \mathsf{sk}_\Theta{}^*_1 \leftarrow (\mathsf{sk}_\Theta, 1), \quad \mathsf{pk}^*_\Theta \leftarrow \mathsf{pk}_\Theta, \quad c^* \leftarrow (1, c_0).$$

Then $\mathsf{KVfy}''_\Theta(\mathsf{sk}_\Theta{}^*_i, \mathsf{pk}^*_\Theta) = 1$ for $i \in \{0,1\}$, but

$$\mathsf{Dec}''_\Theta(\mathsf{sk}_\Theta{}^*_0, c^*) = \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c_0) = m \quad \text{while}$$

$$\mathsf{Dec}''_\Theta(\mathsf{sk}_\Theta{}^*_1, c^*) = \tau\big(\mathsf{Dec}_\Theta(\mathsf{sk}_\Theta, c_0)\big) = \tau(m) \neq m.$$

Hence the experiment returns $1$ with probability $1$, violating key verifiability. □

*Remark 3.* The separation exploits that key verifiability (Fig. 1) quantifies over *arbitrary* ciphertexts $c^*$, whereas decryption soundness (Fig. 2) only checks decryptions of *honestly generated* ciphertexts from $\mathsf{Enc}_\Theta$. Our $\mathsf{Enc}''_\Theta$ treats all honest ciphertexts uniformly ($f = 0$), but allows divergent behavior on malformed ($f = 1$) inputs—sufficient to break key verifiability while preserving decryption soundness.

## C Proofs for the Contrived Construction

### C.1 Proof of Theorem 9 (Trace-Soundness)

**Proof strategy.** We gradually transform the original trace-soundness experiment into a form where all openings are canonical and uniqueness is guaranteed. At each step, either the setup is modified or a consistency check is enforced; any adversary that distinguishes or violates these conditions would immediately break one of the underlying primitives. At the end, the adversary's winning event becomes impossible (up to negligible probability).

- **Game 0 (Original experiment).** The baseline trace-soundness experiment.
- **Game 1 (Simulated CRS for $L_1$).** Replace $\mathsf{crs}^1_\Pi$ by a simulated-CRS. Indistinguishable by zero-knowledge.
- **Game 2 (Simulated CRS for $L_2$).** Replace $\mathsf{crs}^2_\Pi$ analogously. Again indistinguishable by zero-knowledge.
- **Game 3 (Extractable CRS for $L_1$).** Switch to an extractable setup for $\mathsf{crs}^1_\Pi$. Indistinguishable by SSE setup properties.
- **Game 4 (Extractable CRS for $L_2$).** Same for $\mathsf{crs}^2_\Pi$: extractable setup. Again indistinguishable.
- **Game 5 (Extraction from $L_1$).** Use the extractor to check that every accepting $L_1$-proof encodes a consistent ciphertext/signature. Otherwise, simulation-sound extractability would be broken.
- **Game 6 (Extraction from $L_2$).** Extract the secret key from each accepting $L_2$-proof and check decryption consistency. Any violation contradicts simulation-sound extractability.
- **Game 7 (Statement binding).** Abort if a single proof verifies for two distinct statements. This would contradict the statement-binding property of the NIZK.
- **Game 8 (Key-verifiability).** Abort if two functionally different secret keys both pass verification for the same public key. This would break key-verifiability.

After Game 8, two different accepting openings for the same $(m^*, \sigma^*_m)$ are impossible:

- By Game 5, the ciphertext $c^*$ and public key $\mathsf{pk}^*_\Theta$ are fixed.
- By Game 6 and 7, each valid opening corresponds to a unique, consistent decryption.
- By Game 8, there cannot exist two functionally distinct $\mathsf{sk}_\Theta$ for the same $\mathsf{pk}_\Theta$.

Thus the adversary's success probability is negligible, completing the proof.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against trace-soundness. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Original experiment.** The standard trace-soundness experiment. Let $S_0$ be the event that $\mathcal{A}$ outputs $(m^*, \sigma^*_m)$ and two accepting openings to different values as defined.

**Game 1: Simulate $\mathsf{crs}^1_\Pi$.** Generate $(\mathsf{crs}^1_\Pi, \tau^1) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$.

*Transition:* By zk there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \leq \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

$\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)$

1. Receive CRS $\mathsf{crs}_\Pi$.
2. Embed it into the CRS of $\Delta$. The oracles are not needed.
3. Run $\mathcal{A}$ to completion and output its bit. Real-vs-simulated distinguishing advantage transfers to ZK.

**Game 2: Simulate $\mathsf{crs}^2_\Pi$.** Generate $(\mathsf{crs}^2_\Pi, \tau^2) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$.

*Transition:* By ZK there exists $\mathcal{B}^{(2)}_{\mathsf{zk}}$ such that

$$|\Pr[S_2] - \Pr[S_1]| \leq \varepsilon^{(2)}_{\mathsf{zk}}(\lambda).$$

$\mathcal{B}^{(2)}_{\mathsf{zk}}(1^\lambda)$

1. Receive CRS $\mathsf{crs}_\Pi$.
2. Embed it into the CRS of $\Delta$. The oracles are not needed.
3. Run $\mathcal{A}$ and output its bit. Advantage transfers to ZK for $L_2$.

**Game 3: Extractable $\mathsf{crs}^1_\Pi$.** Replace $\mathsf{crs}^1_\Pi$ by an extractable setup $(\mathsf{crs}^1_\Pi, \tau^1, \zeta^1) \leftarrow_\$ \mathcal{E}_1(1^\lambda)$. Keep producing all honest $L_1$ proofs *exactly as in Game 1* (i.e., still via simulation).

*Transition:* By the SSE setup indistinguishability, the distributional change of the CRS is 0:

$$|\Pr[S_3] - \Pr[S_2]| = 0.$$

**Game 4: Extractable $\mathsf{crs}^2_\Pi$.** Replace $\mathsf{crs}^2_\Pi$ by an extractable setup $(\mathsf{crs}^2_\Pi, \tau^2, \zeta^2) \leftarrow_\$ \mathcal{E}_1(1^\lambda)$. Keep producing all honest $L_2$ proofs *exactly as in Game 2* (i.e., still via simulation).

*Transition:* Again by SSE setup indistinguishability,

$$|\Pr[S_4] - \Pr[S_3]| = 0.$$

**Game 5: Extract from $L_1$-proofs (consistency check).** For each accepting $\pi^* \in L_1$ with instance

$$x^* = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, c, m),$$

use the extractor for $\mathsf{crs}^1_\Pi$ to obtain $(r^*, \sigma^*_s, w^*)$ and check

$$c \overset{?}{=} \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, f(w^*); r^*) \quad \wedge \quad \mathsf{Vfy}_\Sigma\big(\mathsf{pk}_\Sigma,\ (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, f(w^*)),\ \sigma^*_s\big) = 1.$$

Abort if extraction fails or any check fails.

*Transition:* If $\mathsf{Vfy}_\Pi(\mathsf{crs}^1_\Pi, x^*, \pi^*) = 1$ but extraction or recomputation fails, SSE is broken. There exists $\mathcal{B}^{(3)}_{\mathsf{sse}}$ with

$$|\Pr[S_5] - \Pr[S_4]| \leq 2\varepsilon^{(3)}_{\mathsf{sse}}(\lambda).$$

$\mathcal{B}^{(3)}_{\mathsf{sse}}(1^\lambda)$

1. Receive the extractable CRS $\mathsf{crs}^1_\Pi$. The oracle is not needed.
2. Run $\mathcal{A}$ in the Game 4 environment; proofs generation is not needed.
3. Upon $\mathcal{A}$ outputting any accepting $(x^*, \pi^*)$ for $L_1$, forward $(x^*, \pi^*)$ to the SSE challenger, which checks extraction/recomputation. Any distinguishing advantage transfers to SSE.

**Game 6: Extract from $L_2$-proofs (consistency check).** For each accepting $\pi_o^* \in L_2$ with instance

$$x_o^* = (\mathsf{pk}_\Sigma^*, \mathsf{pk}_\Theta^*, \mathsf{pk}_\Sigma'^*, c^*, m^*, y^*),$$

extract $\mathsf{sk}_\Theta^*$ and check

$$\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*) = 1 \quad \wedge \quad \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta^*, c^*) = y^*.$$

Abort if extraction fails or recomputation fails.

*Transition:* If $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi^2, x_o^*, \pi_o^*) = 1$ but extraction or recomputation fails, SSE is broken. There exists $\mathcal{B}_{\mathsf{sse}}^{(4)}$ with

$$|\Pr[S_6] - \Pr[S_5]| \leq 2\varepsilon_{\mathsf{sse}}^{(4)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}^{(4)}(1^\lambda)}$

1. Receive the extractable CRS $\mathsf{crs}_\Pi^2$ (the simulation oracle is not needed).
2. Run $\mathcal{A}$ in the Game 5 environment. Upon $\mathcal{A}$ outputting any accepting $(x^*, \pi^*)$ for $L_2$, forward $(x^*, \pi^*)$ to the SSE challenger, which checks extraction/recomputation. Any distinguishing advantage transfers to SSE.

**Game 7: Abort on statement-binding violation.** Abort if a single proof $\pi_o^*$ verifies for two distinct statements $x_0 \neq x_1$ under $\mathsf{crs}_\Pi^2$.

*Transition:* A single proof that verifies for two statements breaks statement-binding. There exists $\mathcal{B}_{\mathsf{sb}}$ with

$$|\Pr[S_7] - \Pr[S_6]| \leq \varepsilon_{\mathsf{sb}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sb}}(1^\lambda)}$

1. Interact with the statement-binding challenger (for the CRS $\mathsf{crs}_\Pi^2$).
2. Run $\mathcal{A}$ in the Game 6 environment.
3. If $\mathcal{A}$ outputs a single $\pi_o^*$ that verifies for two distinct statements $x_0 \neq x_1$, output $(x_0, x_1, \pi_o^*)$ as a valid SB break.

**Game 8: Abort on Key-Verifiability violation.** If two accepting openings for the same $(m^*, \sigma_m^*)$ rely on distinct keys ${\mathsf{sk}_\Theta}_0^* \neq {\mathsf{sk}_\Theta}_1^*$ with both passing $\mathsf{KVfy}_\Theta(\cdot, \mathsf{pk}_\Theta^*) = 1$, abort.

*Transition:* Two distinct valid secret keys for the same $\mathsf{pk}_\Theta$ break key-verifiability. There exists $\mathcal{B}_{\mathsf{kv}}$ such that

$$|\Pr[S_8] - \Pr[S_7]| \leq \varepsilon_{\mathsf{kv}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{kv}}(1^\lambda)}$

1. Interact with the Key-Verifiability challenger; receive $\mathsf{pp}_\Theta$, and embed accordingly.
2. When $\mathcal{A}$ outputs two accepting openings for the same $(m^*, \sigma_m^*)$, we have extracted ${\mathsf{sk}_\Theta}_0^*$ and ${\mathsf{sk}_\Theta}_1^*$ from the respective $L_2$ proofs.
3. Output $({\mathsf{sk}_\Theta}_0^*, {\mathsf{sk}_\Theta}_1^*, \mathsf{pk}_\Theta^*)$ to the KV challenger. If both pass $\mathsf{KVfy}_\Theta(\cdot, \mathsf{pk}_\Theta^*)$ and are distinct, KV is broken.

**Advantage analysis.** By a union bound, we obtain:

$$\Pr[S_0] \leq \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{zk}}^{(2)} + 2\varepsilon_{\mathsf{sse}}^{(3)} + 2\varepsilon_{\mathsf{sse}}^{(4)} + \varepsilon_{\mathsf{sb}} + \varepsilon_{\mathsf{kv}}.$$

All terms are negligible, hence $\Delta$ satisfies trace-soundness as claimed in Theorem 9.

□

**Conclusion.** In summary, any adversary that succeeds in producing two different valid openings for the same $(m^*, \sigma_m^*)$ must necessarily trigger one of the reductions above: either by violating the simulation soundness of $L_1$ or $L_2$, by providing inconsistent witnesses (captured by SSE), by reusing a single proof for two different statements (statement-binding), or by certifying two distinct decryption keys for the same public key (key-verifiability). As each of these underlying assumptions holds with only negligible advantage, we conclude that $\Delta$ satisfies trace-soundness.

### C.2 Proof of Theorem 10 (Non-Frameability)

**Proof strategy.** We again proceed via a sequence of games. Each game gradually modifies the experiment or enforces consistency checks, such that any deviation would contradict one of the underlying assumptions. At the end, the adversary's success event (framing an honest user) becomes impossible.

- **Game 0 (Original experiment).** The standard non-frameability experiment.
- **Game 1 (Simulating $L_1$).** All signature proofs are produced by the ZK simulator. Indistinguishable from real by zero-knowledge.
- **Game 2 (Simulating $L_2$).** All opening proofs are produced by the ZK simulator. Again indistinguishable by zero-knowledge.
- **Game 3 (Extractable CRS for $L_1$).** Switch to an extractable setup for $\mathsf{crs}_\Pi^1$, while still simulating proofs. Indistinguishable by SSE setup indistinguishability.
- **Game 4 (Extractable CRS for $L_2$).** Same for $\mathsf{crs}_\Pi^2$, again indistinguishable.
- **Game 5 (Designated-opening path).** Abort if a forgery succeeds via the designated-opening branch. This would require inverting $f$, breaking one-wayness.
- **Game 6 (Extraction from $L_1$).** Use the extractor on every accepting $L_1$ proof and check consistency. Otherwise, simulation-sound extractability is broken.
- **Game 7 (Witness matches honest identity).** If the extracted witness corresponds to an honest user's identity, we obtain an inversion of $f$ and break one-wayness.
- **Game 8 (Extraction from $L_2$).** Extract $\mathsf{sk}_\Theta^*$ from opening proofs and check decryption consistency. Otherwise, SSE is broken.
- **Game 9 (Bookkeeping).** Extend the signing oracle to record the randomness used in ciphertext generation. This does not change the distribution, but will be needed later.
- **Game 10 (Decryption soundness).** Abort if extraction yields a valid key $\mathsf{sk}_\Theta^*$ but decryption under $\mathsf{sk}_\Theta^*$ does not reproduce the committed value. This would contradict decryption soundness.

After Game 10, the adversary cannot produce a forgery that frames an honest user: - By Games 5 and 7, any designated-opening attempt or witness collision would invert $f$. - By Games 6 and 8, inconsistent NIZK proofs are excluded by SSE. - By Game 10, decryption must be sound.

Thus any successful adversary would break one of the underlying assumptions, and the scheme achieves non-frameability.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against non-frameability. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Original experiment.** The standard non-frameability experiment. Let $S_0$ denote the event that $\mathcal{A}$ succeeds.

**Game 1: Simulate $\mathsf{crs}^1_\Pi$.** Generate $(\mathsf{crs}^1_\Pi, \tau^1) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$ and simulate all signature proofs $\pi$ inside $\sigma_m$.

*Transition:* By ZK there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \leq \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)}$

1. Receive $\mathsf{crs}_\Pi$ and access to a NIZK prover/simulator oracle $\mathcal{O}^{L_1}_{\mathsf{NIZK}}$ from the ZK challenger. The oracle takes $(\mathsf{crs}_\Pi, x; \mathsf{wit})$ and returns a proof $\pi$; whether it simulates or proves is hidden from us.
2. Set $\mathsf{crs}^1_\Pi \leftarrow \mathsf{crs}_\Pi$; prepare the Game 1 environment for $\mathcal{A}$ (all other public parameters as in Game 0).
3. Whenever the environment must produce an honest $L_1$ proof for some instance $x = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, c, m)$: compute the honest witness $(r, \sigma_s, w)$ (we control $r$, the GM-side for $\sigma_s$, and choose $w$ with $f(w) = y$) and query $\pi \leftarrow \mathcal{O}^{L_1}_{\mathsf{NIZK}}(\mathsf{crs}^1_\Pi, x;\ (r, \sigma_s, w))$.
4. Run $\mathcal{A}$ to completion and output its bit. Any distinguishing advantage transfers to ZK for $L_1$.

**Game 2: Simulate $\mathsf{crs}^2_\Pi$.** Generate $(\mathsf{crs}^2_\Pi, \tau^2) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$ and simulate all opening proofs $\pi_o$.

*Transition:* By ZK there exists $\mathcal{B}^{(2)}_{\mathsf{zk}}$ such that

$$|\Pr[S_2] - \Pr[S_1]| \leq \varepsilon^{(2)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(2)}_{\mathsf{zk}}(1^\lambda)}$

1. Receive $\mathsf{crs}_\Pi$ and access to a NIZK prover/simulator oracle $\mathcal{O}^{L_2}_{\mathsf{NIZK}}$ from the ZK challenger. The oracle takes $(\mathsf{crs}_\Pi, x_o; \mathsf{wit})$ and returns a proof $\pi_o$.
2. Set $\mathsf{crs}^2_\Pi \leftarrow \mathsf{crs}_\Pi$; prepare the Game 2 environment for $\mathcal{A}$ (continue as in Game 1 for all other components).
3. Whenever the environment must produce an honest $L_2$ opening proof for $x_o = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, c, m, y)$: use the honest opener witness $\mathsf{sk}_\Theta$ and query $\pi_o \leftarrow \mathcal{O}^{L_2}_{\mathsf{NIZK}}(\mathsf{crs}^2_\Pi, x_o;\ \mathsf{sk}_\Theta)$.
4. Run $\mathcal{A}$ and output its bit. Any distinguishing advantage transfers to ZK for $L_2$.

**Game 3: Extractable $\mathsf{crs}^1_\Pi$.** Replace $\mathsf{crs}^1_\Pi$ by extractable CRS $(\mathsf{crs}^1_\Pi, \tau^1, \zeta^1) \leftarrow_\$ \mathcal{E}(1^\lambda)$.

*Transition:* By SSE we have

$$|\Pr[S_3] - \Pr[S_2]| = 0.$$

**Game 4: Extractable $\mathsf{crs}^2_\Pi$.** Replace $\mathsf{crs}^2_\Pi$ by extractable CRS $(\mathsf{crs}^2_\Pi, \tau^2, \zeta^2) \leftarrow_\$ \mathcal{E}(1^\lambda)$.

*Transition:* By SSE we have

$$|\Pr[S_4] - \Pr[S_3]| = 0.$$

**Game 5: Abort on designated-opening path.** If $\sigma_m$ contains $(\sigma', y')$ with $y' \neq \perp$, then acceptance requires revealing $w$ with $f(w) = y'$. Abort if this path succeeds for an honest identity.

*Transition:* Otherwise OWF is inverted. There exists $\mathcal{B}^{(\mathrm{open})}_{\mathsf{ow}}$ such that

$$|\Pr[S_5] - \Pr[S_4]| \leq \varepsilon^{(\mathrm{open})}_{\mathsf{ow}}(\lambda).$$

$\underline{\mathcal{B}^{(\mathrm{open})}_{\mathsf{ow}}(1^\lambda)}$

1. Receive an OWF challenge $y^* = f(w^*)$.
2. Embed $y^*$ as the public key $\mathsf{upk}^\diamond$ of the fixed honest user. This value is published exactly as in the GS setup.
3. Prepare the Game 4 environment for $\mathcal{A}$: both $\mathsf{crs}^1_\Pi$, $\mathsf{crs}^2_\Pi$ are already set to extractable CRS, and all honest $L_1$ and $L_2$ proofs are simulated as before.
4. Run $\mathcal{A}$ to completion.
5. If $\mathcal{A}$ produces a forgery $\sigma_m$ that contains a designated-opening branch $(\sigma', y')$ with $y' = y^*$ and a valid witness $w$ such that $f(w) = y^*$, output $w$ to the OWF challenger.
6. Otherwise output $\perp$.

**Game 6: Abort on $L_1$ extraction failure / wrong witness.** Extract $(r^*, \sigma_s^*, w^*)$ from $\pi \in L_1$ and check

$$c^* = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, f(w^*); r^*) \quad \wedge \quad \mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, f(w^*)), \sigma_s^*) = 1.$$

Abort on failure.

*Transition:* Otherwise SSE is broken. There exists $\mathcal{B}_{\mathsf{sse}}^{(3)}$ with

$$|\Pr[S_6] - \Pr[S_5]| \leq \varepsilon_{\mathsf{sse}}^{(3)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}^{(3)}(1^\lambda)}$

1. Receive an *extractable* CRS $\hat{\mathsf{crs}}_\Pi$ for $L_1$ from the SSE challenger, and a simulation oracle $\mathsf{SIM}_2^{L_1}$.
2. Set $\mathsf{crs}_\Pi^1 \leftarrow \hat{\mathsf{crs}}_\Pi$. Produce every honest $L_1$ proof via $\mathsf{SIM}_2^{L_1}$: on instance $x$, return $\pi \leftarrow \mathsf{SIM}_2^{L_1}(\mathsf{crs}_\Pi^1, x)$.
3. Run $\mathcal{A}$ to completion, parse $\sigma_m^* = (c^*, \pi^*, \sigma'^*, y'^*)$, set $x^* := (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta^*, \mathsf{pk}'_\Sigma, c^*, m^*)$, and output $(x^*, \pi^*)$ to the SSE challenger.

**Game 7: Abort if extracted $w^*$ matches honest user.** Let $y^\dagger = f(w^*)$. If $y^\dagger = y^\diamond$ for some honest user $y^\diamond = f(w^\diamond)$, abort.

*Transition:* Otherwise OWF is inverted. There exists $\mathcal{B}_{\mathsf{ow}}^{(\mathrm{sigm})}$ with

$$|\Pr[S_7] - \Pr[S_6]| \leq \varepsilon_{\mathsf{ow}}^{(\mathrm{sigm})}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{ow}}^{(\mathrm{sigm})}(1^\lambda)}$

1. Receive an OWF challenge $y^* = f(w^*)$.
2. Locally generate an extractable CRS $(\tilde{\mathsf{crs}}_\Pi{}^1, \tilde{\tau}^1, \tilde{\zeta}^1)$ and embed it as $\mathsf{crs}_\Pi^1$. Honest $L_1$ proofs are produced via simulation as in the previous game (already done).
3. Run $\mathcal{A}$ in the prepared environment.
4. If $\mathcal{A}$ outputs a forgery $(m^*, \sigma_m^*)$ that frames the honest member $\mathsf{upk}^\diamond = y^*$, then from the $L_1$ proof $\pi^*$ in $\sigma_m^*$ extract $(r^*, \sigma_s^*, w^*)$ using $\tilde{\zeta}^1$.
5. If $f(w^*) = y^*$, output $w^*$ as inversion of the OWF challenge. Else output $\perp$.

**Game 8: Abort on $L_2$ extraction failure / wrong witness.** Extract $\mathsf{sk}_\Theta^*$ from $\pi_o \in L_2$ and check

$$\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*) = 1 \quad \wedge \quad \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta^*, c^*) = y^*.$$

Abort otherwise.

*Transition:* Otherwise SSE is broken. There exists $\mathcal{B}_{\mathsf{sse}}^{(4)}$ with

$$|\Pr[S_8] - \Pr[S_7]| \leq \varepsilon_{\mathsf{sse}}^{(4)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}^{(4)}(1^\lambda)}$

1. Receive an extractable CRS $\hat{\mathsf{crs}}_\Pi$ for $L_2$ from the SSE challenger, together with a simulation oracle $\mathsf{SIM}_2^{L_2}$.
2. Set $\mathsf{crs}_\Pi^2 \leftarrow \hat{\mathsf{crs}}_\Pi$. Produce every honest $L_2$ opening proof by querying $\mathsf{SIM}_2^{L_2}$.
3. Run $\mathcal{A}$ to completion in the Game 7 environment.
4. If $\mathcal{A}$ outputs an accepting opening proof $(x_o^*, \pi_o^*) \in L_2$, forward it to the SSE challenger.
5. Any advantage transfers directly to SSE.

**Game 9: Log encryption coins (bookkeeping).** Modify only the signing oracle used to populate the transcript set $\mathcal{Q}$ as follows: whenever the oracle honestly generates a signature $\sigma_m = (c, \pi, \sigma', y')$ with $c \leftarrow \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, y; r)$ for the fixed honest identity $y = f(w)$, it additionally records the randomness $r$ alongside the existing log entry. Formally, instead of inserting $(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m)$ into $\mathcal{Q}$, the oracle inserts $(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, r)$

into an extended log $\mathcal{Q}^{\mathsf{coins}}$ and maintains the natural projection $\mathcal{Q} = \{(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m) : (\cdot, \cdot, \cdot, \sigma_m, \cdot) \in \mathcal{Q}^{\mathsf{coins}}\}$. No other part of the experiment is changed.

*Transition.* The distribution of all values seen by $\mathcal{A}$ is unchanged: the oracle already sampled $r$ uniformly to compute $c = \mathsf{Enc}_{\Theta}(\mathsf{pk}_{\Theta}, y; r)$, and $r$ was never revealed to $\mathcal{A}$. We only record $r$ internally. Hence

$$\left|\Pr[S_9] - \Pr[S_8]\right| = 0.$$

**Game 10: Decryption Soundness.** From Game 9 we have $(\mathsf{sk}^*_{\Theta}, \mathsf{pk}^*_{\Theta}, y^*, r^*)$ with $c^* = \mathsf{Enc}_{\Theta}(\mathsf{pk}^*_{\Theta}, y^*; r^*)$. Abort if $\mathsf{Dec}_{\Theta}(\mathsf{sk}^*_{\Theta}, c^*) \neq y^*$ although $\mathsf{KVfy}_{\Theta}(\mathsf{sk}^*_{\Theta}, \mathsf{pk}^*_{\Theta}) = 1$.

*Transition:* Otherwise DS is violated. There exists $\mathcal{B}_{\mathsf{ds}}$ with

$$|\Pr[S_{10}] - \Pr[S_9]| \leq \varepsilon_{\mathsf{ds}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{ds}}(1^\lambda)}$   1. Receive public parameters $\mathsf{pp}_{\Theta}$ from the DecSound challenger.
    2. Locally generate an extractable CRS $(\tilde{\mathsf{crs}_{\Pi}}^2, \tilde{\tau}^2, \tilde{\zeta}^2)$ for $L_2$ and embed it as $\mathsf{crs}^2_{\Pi}$; produce all honest $L_2$
       proofs via simulation (already done).
    3. Run $\mathcal{A}$ until it outputs a successful forgery $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m, \mathsf{upk}^*, O^*)$.
    4. From the extended log $\mathcal{Q}^{\mathsf{coins}}$, recover $(r^*, \mathsf{pk}^*_{\Theta})$ associated to $\sigma^*_m$. Extract $\mathsf{sk}^*_{\Theta}$ from $O^*$ using $\tilde{\zeta}^2$, and
       set $y^* := \mathsf{upk}^*$.
    5. Output $(\mathsf{sk}^*_{\Theta}, \mathsf{pk}^*_{\Theta}, y^*, r^*)$ to the DecSound challenger.
```

**Advantage analysis.** By a union bound, we obtain:

$$\Pr[S_0] \leq \varepsilon^{(1)}_{\mathsf{zk}} + \varepsilon^{(2)}_{\mathsf{zk}} + \varepsilon^{(\mathrm{open})}_{\mathsf{ow}} + \varepsilon^{(3)}_{\mathsf{sse}} + \varepsilon^{(\mathrm{sigm})}_{\mathsf{ow}} + \varepsilon^{(4)}_{\mathsf{sse}} + \varepsilon_{\mathsf{ds}}.$$

All terms are negligible, hence $\Delta$ satisfies non-frameability as claimed in Theorem 10. □

**Conclusion.** Summarizing, any adversary that successfully frames an honest user must fall into one of the cases captured by the games above: either exploiting the designated-opening path (contradicting the one-wayness of $f$), producing inconsistent $L_1$ or $L_2$ proofs (contradicting simulation-sound extractability), deriving a witness that maps to an honest user identity (contradicting one-wayness of $f$), or violating decryption soundness. As all these underlying assumptions can only be broken with negligible probability, we conclude that $\Delta$ satisfies non-frameability as claimed in Theorem 10.

### C.3 Proof of Theorem 11 (Traceability)

**Proof strategy.** We transform the original traceability experiment step by step. Each game either replaces NIZK proofs by simulated ones, enforces consistency via extraction, or aborts on an impossible forgery path. Any adversary that survives these steps would contradict one of the underlying assumptions. At the end, the adversary cannot win.

- **Game 0 (Original experiment).** Baseline traceability experiment.
- **Game 1 (Simulating $L_1$).** All signature-side NIZK proofs are simulated. By zero-knowledge, indistinguishable from real.
- **Game 2 (Extractable CRS for $L_1$).** Switch $\mathsf{crs}^1_{\Pi}$ to an extractable setup, still simulating proofs. Indistinguishable by SSE setup properties.

- **Game 3 (Extraction from $L_1$).** Extract witnesses from any accepting $L_1$ proof and check ciphertext/signature consistency. Otherwise, simulation-sound extractability is broken.
- **Game 4 (Auxiliary DSIG-forgery path).** Abort if the adversary outputs a non-trivial auxiliary signature that verifies. This would break DSIG unforgeability.

After Game 4, the only way to succeed would be to provide an $L_1$ proof that passes verification but fails extraction, or to forge an auxiliary DSIG. Both are excluded by the underlying assumptions. Thus the adversary's success probability is negligible, proving traceability.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against traceability. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Original experiment.** The standard traceability experiment. Let $S_0$ be the event that $\mathcal{A}$ outputs a verifying signature $\sigma^* = (c^*, \pi^*, \sigma'^*, y'^*)$ on some $m^*$ for which the honest opener fails to output a valid identity.

**Game 1: Simulate $\mathsf{crs}^1_\Pi$.** Generate $(\mathsf{crs}^1_\Pi, \tau^1) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$ and simulate all signature-side NIZK proofs $\pi \in L_1$.

*Transition:* By ZK there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \leq \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)}$

1. Interact with the ZK challenger for $L_1$: receive CRS $\mathsf{crs}_\Pi$ and an oracle $\mathcal{O}^{L_1}_{\mathsf{NIZK}}$.
2. Set $\mathsf{crs}^1_\Pi \leftarrow \mathsf{crs}_\Pi$.
3. For every honest $L_1$ proof inside the signing procedure, query the oracle with the correct witness and return the response.
4. Run $\mathcal{A}$ to completion and output its bit. Any distinguishing advantage transfers to ZK.

**Game 2: Extractable $\mathsf{crs}^1_\Pi$.** Replace $\mathsf{crs}^1_\Pi$ by an extractable CRS $(\mathsf{crs}^1_\Pi, \tau^1, \zeta^1) \leftarrow_\$ \mathcal{E}_1(1^\lambda)$. Honest $L_1$ proofs are still produced exactly as in Game 1 (via simulation).

*Transition:* By SSE setup indistinguishability,

$$|\Pr[S_2] - \Pr[S_1]| = 0.$$

**Game 3: Open via extraction from $L_1$.** Given $\sigma^* = (c^*, \pi^*, \sigma'^*, y'^*)$, define

$$x^* = (\mathsf{pk}_\Sigma, \mathsf{pk}^*_\Theta, \mathsf{pk}'_\Sigma, c^*, m^*).$$

Use the extractor for $\mathsf{crs}^1_\Pi$ to obtain $(r^*, \sigma^*_s, w^*)$ from $\pi^*$. Set $y^\dagger \leftarrow f(w^*)$ and check

$$c^* \stackrel{?}{=} \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, y^\dagger; r^*) \quad \wedge \quad \mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}^*_\Theta, \mathsf{pk}'_\Sigma, y^\dagger), \sigma^*_s) = 1.$$

Abort if extraction or checks fail, otherwise return $y^\dagger$ as the opening.

*Transition:* If $\pi^*$ verifies but extraction/recomputation fails, SSE is broken. There exists $\mathcal{B}^{(2)}_{\mathsf{sse}}$ with

$$|\Pr[S_3] - \Pr[S_2]| \leq \varepsilon^{(2)}_{\mathsf{sse}}(\lambda).$$

$\underline{\mathcal{B}^{(2)}_{\mathsf{sse}}(1^\lambda)}$

1. Receive the extractable CRS $\mathsf{crs}^1_\Pi$ from the SSE challenger (plus simulation oracle if provided).
2. Run $\mathcal{A}$ in the Game 2 environment; produce honest $L_1$ proofs as in Game 2.
3. On $\mathcal{A}$ outputting $(x^*, \pi^*)$, forward to the SSE challenger.
4. If extraction or checks fail, signal violation. Any advantage transfers to SSE.

**Game 4: Abort on auxiliary DSIG-forgery path.** If $\sigma'^* \neq \bot$ and

$$\mathsf{Vfy}_\Sigma(\mathsf{pk}'_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}^*_\Theta, \mathsf{pk}'_\Sigma, y'^*, m^*, c^*, \pi^*), \sigma'^*) = 1,$$

then abort.

*Transition:* Otherwise DSIG unforgeability is violated. There exists $\mathcal{B}_{\mathsf{dsigunf}}$ such that

$$|\Pr[S_4] - \Pr[S_3]| \leq \varepsilon_{\mathsf{dsigunf}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{dsigunf}}(1^\lambda)}$   1. Interact with the $\Sigma$ unforgeability challenger: receive $\mathsf{pk}'_\Sigma$ and access to its signing oracle.
        2. Embed $\mathsf{pk}'_\Sigma$ into the public parameters as the auxiliary key.
        3. Run $\mathcal{A}$ in the Game 3 environment.
        4. If $\mathcal{A}$ outputs $\sigma'^* \neq \bot$ that verifies, forward it as a forgery to the challenger.
```

**Advantage analysis.** By a union bound we obtain

$$\Pr[S_0] \leq \varepsilon^{(1)}_{\mathsf{zk}} + \varepsilon^{(2)}_{\mathsf{sse}} + \varepsilon_{\mathsf{dsigunf}}.$$

All terms are negligible, hence $\Delta$ satisfies traceability as claimed in Theorem 11. □

**Conclusion.** In summary, any adversary that produces a valid signature $\sigma^*$ which cannot be traced to a member must trigger one of the above reductions: either by distinguishing real vs. simulated $L_1$ proofs (breaking ZK), by producing a valid but non-extractable $L_1$ proof (breaking SSE), or by forging an auxiliary DSIG signature (breaking DSIG unforgeability). As each of these assumptions only fails with negligible probability, we conclude that $\Delta$ satisfies traceability.

### C.4 Proof of Theorem 12 (Anonymity)

**Proof strategy.** We transform the anonymity experiment step by step. Each game either checks openability, replaces NIZK proofs by simulated ones, verifies openings by extraction, or switches the challenge encryption. Any distinguishing advantage in these hops would contradict one of the underlying assumptions. At the end, the challenge ciphertext is independent of the hidden bit.

- **Game 0 (Original experiment).** Baseline anonymity experiment.
- **Game 1 (Openability of non-challenge signatures).** The oracle checks that every non-challenge signature can be opened with the honest opener. Otherwise, traceability would be violated.
- **Game 2 (Simulating $L_1$).** All signature-side NIZK proofs are simulated. Indistinguishable by zero-knowledge.
- **Game 3 (Simulating $L_2$).** All opening proofs are simulated. Again indistinguishable by zero-knowledge.
- **Game $4_i$ hybrids (Extraction for $L_2$).** Gradually replace verification of opening proofs by extraction under an extractable CRS. Any gap would contradict simulation-sound extractability.
- **Game 5 (Challenge ciphertext switch).** Replace the challenge ciphertext by an encryption of a fixed dummy value. This hop is indistinguishable by mcINDCPA security of $\Theta$.

After Game 5, the challenge signature is independent of the hidden bit $b$, so $\mathcal{A}$'s success probability is exactly $1/2$. Hence any non-negligible advantage would break one of the underlying assumptions, and the scheme achieves anonymity.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against anonymity. We define the following games and write $S_i$ for the event that $\mathcal{A}$ outputs $b^* = b$ in Game $i$.

**Game 0: Original experiment.** This is the standard anonymity experiment. Let $S_0$ denote the event that $\mathcal{A}$ wins.

**Game 1: Openability check for non-challenge signatures.** Modify the signing oracle so that for every non-challenge query, the challenger runs $\mathsf{Opn}_{\Delta}$ with the honest opener key and aborts if opening fails.

*Transition.* If a non-challenge signature cannot be opened, traceability is broken. There exists $\mathcal{B}_{\mathsf{trace}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \leq \varepsilon_{\mathsf{trace}}(\lambda).$$

**Game 2: Simulate $\mathsf{crs}^1_{\Pi}$.** Generate $(\mathsf{crs}^1_{\Pi}, \tau^1) \leftarrow_{\$} \mathsf{SIM}_1(1^\lambda)$ and simulate all signature-side NIZK proofs.

*Transition.* By ZK there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ with

$$|\Pr[S_2] - \Pr[S_1]| \leq \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)}$ 1. Receive CRS $\mathsf{crs}_{\Pi}$ and access to a NIZK prover/simulator oracle $\mathcal{O}^{L_1}_{\mathsf{NIZK}}$ from the ZK challenger.
2. Set $\mathsf{crs}^1_{\Pi} \leftarrow \mathsf{crs}_{\Pi}$; prepare the Game 2 environment for $\mathcal{A}$ (all other public parameters as in Game 1).
3. Whenever Game 2 needs an honest $L_1$ proof for $x = (\mathsf{pk}_{\Sigma}, \mathsf{pk}_{\Theta}, \mathsf{pk}'_{\Sigma}, c, m)$ with witness $(r, \sigma_s, w)$ (where $f(w) = y$), query $\pi \leftarrow \mathcal{O}^{L_1}_{\mathsf{NIZK}}(\mathsf{crs}^1_{\Pi}, x;\ (r, \sigma_s, w))$.
4. Run $\mathcal{A}$ to completion and output its bit. Any distinguishing advantage transfers to ZK for $L_1$.

**Game 3: Simulate $\mathsf{crs}^2_{\Pi}$.** Generate $(\mathsf{crs}^2_{\Pi}, \tau^2) \leftarrow_{\$} \mathsf{SIM}_1(1^\lambda)$ and simulate all opening proofs.

*Transition.* By ZK there exists $\mathcal{B}^{(2)}_{\mathsf{zk}}$ with

$$|\Pr[S_3] - \Pr[S_2]| \leq \varepsilon^{(2)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{zk}}}^{(2)}(1^\lambda)$ 1. Receive CRS $\mathsf{crs}_{\Pi}$ and access to a NIZK prover/simulator oracle $\mathcal{O}^{L_2}_{\mathsf{NIZK}}$ from the ZK challenger.
2. Set $\mathsf{crs}^2_{\Pi} \leftarrow \mathsf{crs}_{\Pi}$; prepare the Game 3 environment for $\mathcal{A}$ (continue as in Game 2 for all other components).
3. Whenever Game 3 needs an honest $L_2$ proof for $x_o = (\mathsf{pk}_{\Sigma}, \mathsf{pk}_{\Theta}, \mathsf{pk}'_{\Sigma}, c, m, y)$ with witness $\mathsf{sk}_{\Theta}$, query $\pi_o \leftarrow \mathcal{O}^{L_2}_{\mathsf{NIZK}}(\mathsf{crs}^2_{\Pi}, x_o;\ \mathsf{sk}_{\Theta})$.
4. Run $\mathcal{A}$ and output its bit. Any distinguishing advantage transfers to ZK for $L_2$.

**Game 4$_i$ hybrids for $0 \leq i \leq q_o$.** For the first $i$ (non-challenge) opening queries, the challenger verifies by extraction under an extractable CRS for $\mathsf{crs}^2_{\Pi}$: for each accepting $\pi_o \in L_2$ on $x = (\mathsf{pk}_{\Theta}, c, y)$ extract $\mathsf{sk}_{\Theta}$ and check $\mathsf{KVfy}_{\Theta}(\mathsf{sk}_{\Theta}, \mathsf{pk}_{\Theta}) = 1$ and $\mathsf{Dec}_{\Theta}(\mathsf{sk}_{\Theta}, c) = y$. Remaining queries are checked by $\mathsf{Vfy}_{\Pi}$. Game $4_0$=Game 3 and Game $4_{q_o}$=Game 4.

*Transition.* If the $i$-th query distinguishes, then a proof verifies but extraction fails or recomputation fails, contradicting SSE. Hence there exists $\mathcal{B}^{(2,i)}_{\mathsf{sse}}$ such that

$$|\Pr[S^{(i)}_4] - \Pr[S^{(i-1)}_4]| \leq \varepsilon^{(2)}_{\mathsf{sse}}(\lambda).$$

**Game 5: Replace the challenge encryption.** In the challenge signing oracle, replace $c^*$ by an encryption of a fixed dummy value, $c^* \leftarrow \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, 0)$. All NIZKs remain simulated as before.

*Transition.* This hop is indistinguishable by mcINDCPA of $\Theta$. There exists $\mathcal{B}_{\mathsf{mcindcpa}}$ with

$$|\Pr[S_5] - \Pr[S_4^{(q_o)}]| \leq \varepsilon_{\mathsf{mcindcpa}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{mcindcpa}}(1^\lambda)}$
  1. Receive $\mathsf{pp}_\Theta$ and $\mathsf{pk}_\Theta$ from the mcINDCPA challenger, with LR oracle access.
  2. In the challenge signing query, submit $(f(w_0), f(w_1))$ as LR messages and receive the challenge ciphertext $c^*$.
  3. Embed $c^*$ into $\sigma_m^*$ and continue Game 5 with $\mathcal{A}$.
  4. When $\mathcal{A}$ outputs guess $b^*$, forward it to the mcINDCPA challenger.
```

**Advantage analysis.** Let $\varepsilon(\lambda) = \Pr[S_0] - \frac{1}{2}$. Then

$$|\varepsilon(\lambda)| \leq \varepsilon_{\mathsf{trace}} + \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{zk}}^{(2)} + q_o \cdot \varepsilon_{\mathsf{sse}}^{(2)} + \varepsilon_{\mathsf{mcindcpa}},$$

which is negligible. □

**Conclusion.** If $\mathcal{A}$ distinguishes the challenge bit with non-negligible advantage, then along the game hops we obtain a distinguisher for traceability (Game 0→1), or for ZK of $L_1$ or $L_2$ (Games 1→2, 2→3), or for SSE of $L_2$ (hybrids in Game 4), or an mcINDCPA adversary against $\Theta$ (Game 4→5). Since all underlying advantages are negligible, anonymity holds.

### C.5 Proof of Theorem 13 (Non-Claimability)

**Proof strategy.** We transform the non-claimability experiment step by step. Each game either replaces NIZK proofs by simulated ones, enforces consistency via extraction, or checks decryption soundness. Any adversary that survives these steps would contradict one of the underlying assumptions. At the end, the adversary cannot claim an honestly issued signature as its own.

- **Game 0 (Original experiment).** Baseline non-claimability experiment. Logged signatures always have the designated-verifier branch inert.
- **Game 1 (Simulating $L_2$).** All opening proofs are produced by the ZK simulator. Indistinguishable from real by zero-knowledge.
- **Game 2 (Extractable CRS for $L_2$).** Switch to an extractable setup, still simulating proofs. Indistinguishable by SSE setup properties.
- **Game 3 (Extraction from $L_2$).** Extract secret keys from accepting $L_2$ proofs and check recomputation. Otherwise, simulation-sound extractability is broken.
- **Game 4 (Decryption soundness).** Verify that any extracted secret key decrypts honestly generated ciphertexts correctly. Otherwise, Decryption Soundness is broken.

After Game 4, no adversary can output a claim on a transcript from the log that opens to a different member identity. Any such attempt would contradict ZK, SSE, or Decryption Soundness. Thus the scheme is non-claimable.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against non-claimability. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Real experiment.** This is $\mathbf{Exp}^{\mathsf{NonClaimability}}_{\mathcal{A},\Delta}(\lambda)$ from Fig. 8. By definition of the oracle $\mathcal{O}_{\mathsf{Sgn}_\Delta}$, every logged signature

$$(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m) \in \mathcal{Q} \quad \text{has} \quad \sigma_m = (c, \pi, \bot, \bot).$$

Hence, when the experiment later checks $\mathsf{Jdg}_\Delta$ on a transcript from $\mathcal{Q}$, the designated-verifier branch in $\mathsf{Vfy}_\Delta/\mathsf{Jdg}_\Delta$ is inert. Any attempt by $\mathcal{A}$ to modify $(m, \sigma_m)$ to re-enable that branch necessarily leaves $\mathcal{Q}$ and is rejected.

**Game 1: Simulate $\mathsf{crs}^2_\Pi$.** Generate $(\mathsf{crs}^2_\Pi, \tau^2) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$ and simulate all $L_2$ proofs.

*Transition:* By ZK there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \le \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)}$ 1. Receive $\mathsf{crs}_\Pi$ and access to a NIZK oracle $\mathcal{O}^{L_2}_{\mathsf{NIZK}}$ from the ZK challenger.
2. Set $\mathsf{crs}^2_\Pi \leftarrow \mathsf{crs}_\Pi$. For every honest $L_2$ proof on instance $x_o = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, \mathsf{pk}'_\Sigma, c, m, y)$, query $\mathcal{O}^{L_2}_{\mathsf{NIZK}}(\mathsf{crs}^2_\Pi, x_o; \mathsf{sk}_\Theta)$ with the honest $\mathsf{sk}_\Theta$ and return the result.
3. Run $\mathcal{A}$ and output its bit. Any distinguishing advantage transfers to ZK.

**Game 2: Extractable $\mathsf{crs}^2_\Pi$.** Replace $\mathsf{crs}^2_\Pi$ by an extractable CRS $(\mathsf{crs}^2_\Pi, \tau^2, \zeta^2) \leftarrow_\$ \mathcal{E}_1(1^\lambda)$.

*Transition:* By SSE we have

$$|\Pr[S_2] - \Pr[S_1]| = 0.$$

**Game 3: Abort on $L_2$ extraction failure / wrong witness.** When $\mathcal{A}$ outputs $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m, \mathsf{upk}^*, O^*)$ with $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m) \in \mathcal{Q}$ and $\mathsf{upk}^* \neq \mathsf{upk}$, parse $\sigma^*_m = (c^*, \pi^*, \bot, \bot)$ and $O^* = \pi^*_o$. Form $x^*_o = (\mathsf{pk}^*_\Sigma, \mathsf{pk}^*_\Theta, \mathsf{pk}'_\Sigma{}^*, c^*, m^*, y^*)$ with $y^* = \mathsf{upk}^*$ and run the extractor to obtain $\mathsf{sk}^*_\Theta$. Check $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1$ and $\mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c^*) = y^*$. Abort if extraction or recomputation fails.

*Transition:* Otherwise SSE is broken. There exists $\mathcal{B}^{(2)}_{\mathsf{sse}}$ such that

$$|\Pr[S_3] - \Pr[S_2]| \le \varepsilon^{(2)}_{\mathsf{sse}}(\lambda).$$

$\underline{\mathcal{B}^{(2)}_{\mathsf{sse}}(1^\lambda)}$ 1. Receive an extractable CRS $\hat{\mathsf{crs}}_\Pi$ for $L_2$ and a simulation oracle $\mathsf{SIM}^{L_2}_2$ from the SSE challenger.
2. Set $\mathsf{crs}^2_\Pi \leftarrow \hat{\mathsf{crs}}_\Pi$. For every honest $L_2$ proof on instance $x_o$, query $\mathsf{SIM}^{L_2}_2(\mathsf{crs}^2_\Pi, x_o)$.
3. Run $\mathcal{A}$ to completion and obtain its final pair $(x^*_o, \pi^*_o)$; output it to the SSE challenger.

**Game 4: Abort on DS violation.** Since $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m) \in \mathcal{Q}$, the ciphertext $c^*$ was honestly generated as $c^* = \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, y; r^*)$ for the fixed honest $y = f(w)$ with coins $r^*$ known to the challenger from the signing oracle's internal state. Abort if $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1$ but $\mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c^*) \neq y$.

*Transition:* Otherwise Decryption Soundness is broken. There exists $\mathcal{B}_{\mathsf{ds}}$ such that

$$|\Pr[S_4] - \Pr[S_3]| \le \varepsilon_{\mathsf{ds}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{ds}}(1^\lambda)}$   1. Receive $\mathsf{pp}_\Theta$ from the DS challenger.
    2. Run $\mathcal{A}$ in the Game 3 environment until it outputs $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, \mathsf{upk}^*, O^*)$.
    3. From the experiment's internal record recover the coins $r^*$ used in $c^* = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta^*, y; r^*)$.
    4. From $O^*$ extract $\mathsf{sk}_\Theta^*$ consistent with $\mathsf{pk}_\Theta^*$ and the claimed $y^* = \mathsf{upk}^*$.
    5. Output $(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*, y, r^*)$ to the DS challenger. Any non-negligible success breaks Decryption Soundness.
```

**Advantage analysis.**

$$\Pr[S_0] \leq \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{sse}}^{(2)} + \varepsilon_{\mathsf{ds}},$$

which is negligible. Therefore $\Delta$ is non-claimable. □

**Conclusion.** A successful non-claimability adversary must either distinguish real vs. simulated $L_2$ proofs (violating ZK), produce an $L_2$ proof that verifies without a consistent witness (violating SSE), or present a valid decryption key that fails to decrypt an honestly formed ciphertext to its plaintext (violating Decryption Soundness). Since each underlying advantage is negligible, $\Delta$ satisfies non-claimability.

## D Proofs for the Secure Scheme

### D.1 Proof of Theorem 15 (Trace-Soundness)

**Proof strategy.** We gradually transform the trace-soundness experiment into one where all openings are enforced by extraction and consistency checks. Each hop either changes the CRS distribution, activates extraction, or aborts if the adversary violates a binding property. Any adversary that distinguishes along the way would break one of the underlying assumptions. At the end, two different valid openings for the same $(m^*, \sigma_m^*)$ are impossible.

- **Game 0 (Original experiment).** Baseline trace-soundness experiment.
- **Game 1 (Simulated CRS for $L_1$).** Replace $\mathsf{crs}_\Pi^1$ by a simulated-CRS. Indistinguishable by zero-knowledge.
- **Game 2 (Simulated CRS for $L_2$).** Replace $\mathsf{crs}_\Pi^2$ analogously. Again indistinguishable by zero-knowledge.
- **Game 3 (Extractable CRS for $L_1$).** Switch $\mathsf{crs}_\Pi^1$ to an extractable setup. Distribution preserved by SSE.
- **Game 4 (Extractable CRS for $L_2$).** Same for $\mathsf{crs}_\Pi^2$. Distribution preserved.
- **Game 5 (Extraction from $L_1$).** Check that any accepting $L_1$ proof extracts to a consistent ciphertext/signature. Otherwise, SSE is broken.
- **Game 6 (Extraction from $L_2$).** Check that any accepting $L_2$ proof extracts to a consistent secret key. Otherwise, SSE is broken.
- **Game 7 (Statement binding).** Abort if one proof verifies for two different statements. Otherwise statement-binding is violated.
- **Game 8 (Key-verifiability).** Abort if two distinct secret keys verify for the same public key. Otherwise key-verifiability is broken.

After Game 8, every accepting opening is consistent and unique: SSE excludes inconsistent witnesses, statement-binding excludes re-use of a single proof for two statements, and key-verifiability excludes two distinct keys for the same $\mathsf{pk}_\Theta$. Thus double openings are impossible and the scheme is trace-sound.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against trace-soundness. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Original experiment.** The standard trace-soundness experiment (Fig. 7). Let $S_0$ be the event that $\mathcal{A}$ outputs $(m^*, \sigma_m^*)$ and two accepting openings to *different* identities for the same pair $(m^*, \sigma_m^*)$.

**Game 1: Replace $\mathsf{crs}_\Pi^1$ by a simulated-CRS.** Sample $(\mathsf{crs}_\Pi^1, \tau^1) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$. No further changes; we do not generate any honest $L_1$ proofs in this experiment.

*Transition.* CRS-distribution hop only. By ZK there exists $\mathcal{B}_{\mathsf{zk}}^{(1)}$ with

$$|\Pr[S_1] - \Pr[S_0]| \le \varepsilon_{\mathsf{zk}}^{(1)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{zk}}^{(1)}(1^\lambda)}$
1. Receive a CRS $\mathsf{crs}_\Pi$ from the ZK challenger (real-or-simulated, hidden).
2. Set $\mathsf{crs}_\Pi^1 \leftarrow \mathsf{crs}_\Pi$; keep all other components as in Game 0.
3. Run $\mathcal{A}$ to completion and output 1 iff $\mathcal{A}$ wins. Any non-negligible gap distinguishes the CRS distribution.

**Game 2: Replace $\mathsf{crs}_\Pi^2$ by a simulated-CRS.** Sample $(\mathsf{crs}_\Pi^2, \tau^2) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$. No further changes.

*Transition.* CRS-distribution hop only. By ZK there exists $\mathcal{B}_{\mathsf{zk}}^{(2)}$ with

$$|\Pr[S_2] - \Pr[S_1]| \le \varepsilon_{\mathsf{zk}}^{(2)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{zk}}^{(2)}(1^\lambda)}$
1. Receive a CRS $\mathsf{crs}_\Pi$ from the ZK challenger.
2. Set $\mathsf{crs}_\Pi^2 \leftarrow \mathsf{crs}_\Pi$; environment as in Game 1.
3. Run $\mathcal{A}$; output 1 iff $\mathcal{A}$ wins.

**Game 3: Make $\mathsf{crs}_\Pi^1$ extractable.** Replace $\mathsf{crs}_\Pi^1$ by $(\mathsf{crs}_\Pi^1, \tau^1, \zeta^1) \leftarrow_\$ \mathcal{E}(1^\lambda)$. Environment otherwise unchanged.

*Transition.* By SSE the setup distribution is preserved (per definition), hence

$$|\Pr[S_3] - \Pr[S_2]| = 0.$$

**Game 4: Make $\mathsf{crs}_\Pi^2$ extractable.** Replace $\mathsf{crs}_\Pi^2$ by $(\mathsf{crs}_\Pi^2, \tau^2, \zeta^2) \leftarrow_\$ \mathcal{E}(1^\lambda)$.

*Transition.* By SSE the setup distribution is preserved, hence

$$|\Pr[S_4] - \Pr[S_3]| = 0.$$

**Game 5: Extract from $L_1$-proofs.** For each accepting $\pi^* \in L_1$ with instance $x^* = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m)$, extract $(r^*, \sigma_s^*, w^*)$ and check

$$c \stackrel{?}{=} \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, f(w^*); r^*), \qquad \mathsf{Vfy}_\Sigma\big(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, f(w^*)), \sigma_s^*\big) = 1.$$

Abort if extraction fails or any check fails.

*Transition.* If $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi^1, x^*, \pi^*) = 1$ but extraction/recomputation fails, SSE is broken. There exists $\mathcal{B}_{\mathsf{sse}}^{(3)}$ with

$$|\Pr[S_5] - \Pr[S_4]| \le 2\varepsilon_{\mathsf{sse}}^{(3)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}^{(3)}(1^\lambda)}$
1. Fix $\mathsf{crs}_\Pi^1$ to an extractable CRS as in Game 3.
2. Run $\mathcal{A}$; whenever $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi^1, x^*, \pi^*) = 1$ for $L_1$, forward $(x^*, \pi^*)$ to the SSE challenger.

**Game 6: Extract from $L_2$-proofs.** For each accepting $\pi_o^* \in L_2$ with instance $x_o^* = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta^*, c^*, m, y^*)$, extract $\mathsf{sk}_\Theta^*$ and check

$$\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*) = 1, \qquad \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta^*, c^*) = y^*.$$

Abort if extraction fails or recomputation fails.

*Transition.* If $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi^2, x_o^*, \pi_o^*) = 1$ but extraction/recomputation fails, SSE is broken. There exists $\mathcal{B}_{\mathsf{sse}}^{(4)}$ with

$$|\Pr[S_6] - \Pr[S_5]| \leq 2\varepsilon_{\mathsf{sse}}^{(4)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}^{(4)}(1^\lambda)}$ 1. Fix $\mathsf{crs}_\Pi^2$ to an extractable CRS as in Game 4.
2. Run $\mathcal{A}$; whenever $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi^2, x_o^*, \pi_o^*) = 1$ for $L_2$, forward $(x_o^*, \pi_o^*)$ to the SSE challenger.

**Game 7: Abort on statement-binding violation.** Abort if one $\pi_o^*$ verifies under $\mathsf{crs}_\Pi^2$ for two distinct statements $x_0 \neq x_1$.

*Transition.* Such a $\pi_o^*$ breaks statement-binding. There exists $\mathcal{B}_{\mathsf{sb}}$ with

$$|\Pr[S_7] - \Pr[S_6]| \leq \varepsilon_{\mathsf{sb}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sb}}(1^\lambda)}$ 1. Fix $\mathsf{crs}_\Pi^2$ to the CRS provided by the SB challenger.
2. Run $\mathcal{A}$; if one $\pi_o^*$ verifies for two distinct statements $x_0 \neq x_1$, output $(x_0, x_1, \pi_o^*)$.

**Game 8: Abort on key-verifiability violation.** If two accepting openings for the same $(m^*, \sigma_m^*)$ rely on distinct keys $\mathsf{sk}_{\Theta}{}_0^* \neq \mathsf{sk}_{\Theta}{}_1^*$ that both satisfy $\mathsf{KVfy}_\Theta(\cdot, \mathsf{pk}_\Theta^*) = 1$, abort.

*Transition.* Two distinct valid keys for the same $\mathsf{pk}_\Theta^*$ break key-verifiability. There exists $\mathcal{B}_{\mathsf{kv}}$ such that

$$|\Pr[S_8] - \Pr[S_7]| \leq \varepsilon_{\mathsf{kv}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{kv}}(1^\lambda)}$ 1. Receive $\mathsf{pp}_\Theta$ from the KV challenger. Embed it accordingly.
2. Run $\mathcal{A}$ to obtain two accepting openings for the same $(m^*, \sigma_m^*)$ (by assumption).
3. Extract $\mathsf{sk}_{\Theta}{}_0^*, \mathsf{sk}_{\Theta}{}_1^*$ from the respective $L_2$ proofs and output $(\mathsf{sk}_{\Theta}{}_0^*, \mathsf{sk}_{\Theta}{}_1^*, \mathsf{pk}_\Theta^*)$ to the KV challenger.

**Advantage analysis.**

$$\Pr[S_0] \leq \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{zk}}^{(2)} + 2\varepsilon_{\mathsf{sse}}^{(3)} + 2\varepsilon_{\mathsf{sse}}^{(4)} + \varepsilon_{\mathsf{sb}} + \varepsilon_{\mathsf{kv}}.$$

All negligible, so $\Delta'$ is trace-sound. □

**Conclusion.** A successful adversary producing two different accepting openings for a single $(m^*, \sigma_m^*)$ must either make an $L_1$ or $L_2$ proof verify without a consistent witness (breaking SSE), reuse a single $L_2$ proof for two distinct statements (breaking statement-binding), or certify two distinct secret keys for the same public key (breaking key-verifiability). Since each event has negligible probability, $\Delta'$ satisfies trace-soundness.

### D.2 Proof of Theorem 16 (Non-Frameability)

**Proof strategy.** We gradually transform the non-frameability experiment into a form where any successful forgery would contradict one of the underlying primitives. Each hop either replaces CRS by simulated or extractable ones, enforces consistency of extracted witnesses, or checks that extracted values cannot collide with an honest user or violate decryption soundness.

- **Game 0 (Original experiment).** Baseline non-frameability experiment.
- **Game 1 (Simulated $L_1$).** Replace $\mathsf{crs}^1_\Pi$ by a simulated CRS; honest $L_1$ proofs via ZK oracle. Indistinguishable by zero-knowledge.
- **Game 2 (Simulated $L_2$).** Replace $\mathsf{crs}^2_\Pi$ analogously. Indistinguishable by zero-knowledge.
- **Game 3 (Extractable $L_1$).** Switch $\mathsf{crs}^1_\Pi$ to an extractable CRS. Distribution preserved by SSE.
- **Game 4 (Extractable $L_2$).** Switch $\mathsf{crs}^2_\Pi$ analogously. Distribution preserved.
- **Game 5 (Extraction from $L_1$).** Check that accepting $L_1$ proofs extract to a consistent ciphertext/signature. Otherwise SSE is broken.
- **Game 6 (Extraction from $L_2$).** Check that accepting $L_2$ proofs extract to a consistent secret key/decryption. Otherwise SSE is broken.
- **Game 7 (OWF check).** Abort if the extracted witness corresponds to an honest member's identity. If so, one-wayness of $f$ is broken.
- **Game 8 (Decryption soundness).** Check that extracted values decrypt honestly generated ciphertexts correctly. Otherwise Decryption Soundness is broken.

After Game 8, any forgery that frames the honest member would contradict ZK, SSE, OWF, or Decryption Soundness. Thus the scheme is non-frameable.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against non-frameability. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Original experiment.** The standard non-frameability experiment (Fig. 6). Let $S_0$ be the event that $\mathcal{A}$ outputs $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m, O^*)$ with $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m) \notin \mathcal{Q}$ and $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma^*_m, \mathsf{upk}, O^*) = 1$, where $\mathsf{upk}$ is the fixed honest member's public key.

**Game 1: Simulated $\mathsf{crs}^1_\Pi$.** Generate $(\mathsf{crs}^1_\Pi, \tau^1) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$. Use this CRS for all $L_1$ verifications; the environment is otherwise unchanged (honest $L_1$-proofs in $\mathcal{O}_{\mathsf{Sgn}_\Delta}$ are answered via the ZK oracle).

*Transition.* By ZK there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \leq \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

```
$\underline{\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)}$   1. Receive $\mathsf{crs}_\Pi$ and access to a prover/simulator oracle $\mathcal{O}^{L_1}_{\mathsf{NIZK}}$ for $L_1$ from the ZK challenger.
    2. Set $\mathsf{crs}^1_\Pi \leftarrow \mathsf{crs}_\Pi$; prepare the Game 1 environment.
    3. Whenever $\mathcal{O}_{\mathsf{Sgn}_\Delta}$ must produce an honest $L_1$ proof for $x = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m)$ with witness $(r, \sigma_s, w)$,
       query $\pi \leftarrow \mathcal{O}^{L_1}_{\mathsf{NIZK}}(\mathsf{crs}^1_\Pi, x;\ (r, \sigma_s, w))$ and return $\pi$.
    4. Run $\mathcal{A}$ to completion and output its bit. Any distinguishing advantage transfers to ZK.
```

**Game 2: Simulated $\mathsf{crs}^2_\Pi$.** Generate $(\mathsf{crs}^2_\Pi, \tau^2) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$. No further changes (no honest $L_2$-proofs are produced in this experiment).

*Transition.* By ZK there exists $\mathcal{B}^{(2)}_{\mathsf{zk}}$ such that

$$|\Pr[S_2] - \Pr[S_1]| \leq \varepsilon^{(2)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(2)}_{\mathsf{zk}}(1^\lambda)}$

1. Receive a CRS $\mathsf{crs}_\Pi$ from the ZK challenger for $L_2$ (real-or-simulated, hidden).
2. Set $\mathsf{crs}^2_\Pi \leftarrow \mathsf{crs}_\Pi$; keep the rest as in Game 1 (no honest $L_2$ proofs are needed).
3. Run $\mathcal{A}$ and output 1 iff $\mathcal{A}$ wins. Any non-negligible gap distinguishes the CRS distribution.

**Game 3: Extractable $\mathsf{crs}^1_\Pi$.** Replace $\mathsf{crs}^1_\Pi$ by $(\mathsf{crs}^1_\Pi, \tau^1, \zeta^1) \leftarrow_\$ \mathcal{E}(1^\lambda)$. Honest proofs are still produced via the ZK oracle.

*Transition.* By SSE setup indistinguishability,

$$|\Pr[S_3] - \Pr[S_2]| = 0.$$

**Game 4: Extractable $\mathsf{crs}^2_\Pi$.** Replace $\mathsf{crs}^2_\Pi$ by $(\mathsf{crs}^2_\Pi, \tau^2, \zeta^2) \leftarrow_\$ \mathcal{E}(1^\lambda)$.

*Transition.* By SSE,

$$|\Pr[S_4] - \Pr[S_3]| = 0.$$

**Game 5: Abort on $L_1$ extraction failure / wrong witness.** For an accepting output with $\sigma^*_m = (c^*, \pi^*)$ on $x^* = (\mathsf{pk}^*_\Sigma, \mathsf{pk}^*_\Theta, c^*, m^*)$, extract $(r^*, \sigma^*_s, w^*)$ and check

$$c^* \overset{?}{=} \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, f(w^*); r^*) \quad \wedge \quad \mathsf{Vfy}_\Sigma(\mathsf{pk}^*_\Sigma, (\mathsf{pk}^*_\Sigma, \mathsf{pk}^*_\Theta, f(w^*)), \sigma^*_s) = 1.$$

Abort if extraction or recomputation fails.

*Transition.* Otherwise SSE is broken. There exists $\mathcal{B}^{(3)}_{\mathsf{sse}}$ with

$$|\Pr[S_5] - \Pr[S_4]| \leq \varepsilon^{(3)}_{\mathsf{sse}}(\lambda).$$

$\underline{\mathcal{B}^{(3)}_{\mathsf{sse}}(1^\lambda)}$

1. Fix $\mathsf{crs}^1_\Pi$ to an extractable CRS as in Game 4 (simulation capability for honest $L_1$ proofs is available via the oracle).
2. Run $\mathcal{A}$ in this environment; whenever $\mathsf{Vfy}_\Pi(\mathsf{crs}^1_\Pi, x^*, \pi^*) = 1$ for $L_1$, forward $(x^*, \pi^*)$ to the SSE challenger.

**Game 6: Abort on $L_2$ extraction failure / wrong witness.** From $O^* = \pi^*_o$ extract $\mathsf{sk}^*_\Theta$ on instance $x^*_o = (\mathsf{pk}^*_\Sigma, \mathsf{pk}^*_\Theta, c^*, m^*, y^*)$ with $y^* = \mathsf{upk}$. Check

$$\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1 \quad \wedge \quad \mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c^*) = y^*.$$

Abort if extraction or recomputation fails.

*Transition.* Otherwise SSE is broken. There exists $\mathcal{B}^{(4)}_{\mathsf{sse}}$ with

$$|\Pr[S_6] - \Pr[S_5]| \leq \varepsilon^{(4)}_{\mathsf{sse}}(\lambda).$$

$\underline{\mathcal{B}^{(4)}_{\mathsf{sse}}(1^\lambda)}$

1. Fix $\mathsf{crs}^2_\Pi$ to an extractable CRS as in Game 5.
2. Run $\mathcal{A}$; when $\mathsf{Vfy}_\Pi(\mathsf{crs}^2_\Pi, x^*_o, \pi^*_o) = 1$ for $L_2$, forward $(x^*_o, \pi^*_o)$ to the SSE challenger.

**Game 7: Abort if extracted $w^*$ matches honest user.** Let $y^\dagger = f(w^*)$. If $y^\dagger = y^\diamond$ for some honest user $y^\diamond = f(w^\diamond)$, abort.

*Transition.* Otherwise OWF is inverted. There exists $\mathcal{B}_{\mathsf{ow}}^{(\mathrm{sigm})}$ with

$$|\Pr[S_7] - \Pr[S_6]| \leq \varepsilon_{\mathsf{ow}}^{(\mathrm{sigm})}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{ow}}^{(\mathrm{sigm})}(1^\lambda)}$

1. Receive an OWF challenge $y^* = f(w^*)$.
2. Locally set $\mathsf{crs}_\Pi^1$ to an extractable CRS (as allowed by the OWF-to-SSE embedding) while keeping the environment identical to Game 6.
3. Run $\mathcal{A}$; on success that frames $\mathsf{upk}^\diamond = y^*$, extract $(r^*, \sigma_s^*, w^*)$ from the verifying $L_1$ proof $\pi^*$.
4. If $f(w^*) = y^*$, output $w^*$ to the OWF challenger; else output $\perp$.

**Game 8: Decryption-Soundness consistency.** With the extracted values, $c^* = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta^*, f(w^*); r^*)$ and $\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*) = 1$. Abort if $\mathsf{Dec}_\Theta(\mathsf{sk}_\Theta^*, c^*) \neq f(w^*)$.

*Transition.* Otherwise DS is violated. There exists $\mathcal{B}_{\mathsf{ds}}$ with

$$|\Pr[S_8] - \Pr[S_7]| \leq \varepsilon_{\mathsf{ds}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{ds}}(1^\lambda)}$

1. Use the tuple $(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*, y, f(w^*), r^*)$ obtained in Game 7, where $y = f(w^*)$ and $c^* = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta^*, y; r^*)$.
2. Output $(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*, y, r^*)$ to the DecSound challenger.

**Advantage analysis.** By a union bound we obtain

$$\Pr[S_0] \leq \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{zk}}^{(2)} + \varepsilon_{\mathsf{sse}}^{(3)} + \varepsilon_{\mathsf{sse}}^{(4)} + \varepsilon_{\mathsf{ow}}^{(\mathrm{sigm})} + \varepsilon_{\mathsf{ds}}.$$

All terms are negligible, hence $\Delta'$ is non-frameable as claimed. □

**Conclusion.** In summary, any adversary that succeeds in framing the honest user must necessarily exploit one of the following: producing inconsistent $L_1$ or $L_2$ proofs (contradicting SSE), deriving a witness that maps to the honest identity (contradicting OWF), or violating decryption soundness. As each of these underlying assumptions holds only with negligible probability, we conclude that $\Delta'$ satisfies non-frameability.

### D.3 Proof of Theorem 17 (Traceability)

**Proof strategy.** We transform the traceability experiment step by step. Each hop either replaces the CRS by a simulated one or enforces extraction to guarantee that any verifying proof encodes a consistent ciphertext/signature. Any adversary that distinguishes along the way must break ZK or SSE. At the end, the honest opener always recovers a valid identity.

- **Game 0 (Original experiment).** Baseline traceability experiment: adversary wins if it outputs a verifying signature that the opener cannot link to a member.
- **Game 1 (Simulated $L_1$).** Replace $\mathsf{crs}_\Pi^1$ by a simulated CRS and produce all $L_1$ proofs via the ZK oracle. Indistinguishable by zero-knowledge.
- **Game 2 (Extractable $L_1$).** Switch $\mathsf{crs}_\Pi^1$ to an extractable CRS, proofs still simulated. Distribution preserved by SSE.

- **Game 3 (Extraction from $L_1$).** Extract witnesses from any accepting $L_1$ proof and check recomputation of ciphertext and DSIG. Otherwise, simulation-sound extractability is broken.

After Game 3, every verifying signature can be opened consistently by extraction, so the opener always recovers a valid identity. Thus any adversary winning traceability would contradict ZK or SSE, and the scheme is traceable.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against traceability. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Original experiment.** The standard traceability experiment (Fig. 5). Let $S_0$ be the event that $\mathcal{A}$ outputs a verifying signature $\sigma^* = (c^*, \pi^*)$ on $m^*$ such that $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m^*, \sigma^*) = 1$ but the honest opener fails to recover a valid member identity.

**Game 1: Simulated $\mathsf{crs}^1_\Pi$.** Generate $(\mathsf{crs}^1_\Pi, \tau^1) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$ and simulate all signature-side NIZK proofs $\pi \in L_1$. All other parts of the experiment are unchanged.

*Transition.* By ZK there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \le \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

```
$\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)$
  1. Receive CRS $\mathsf{crs}_\Pi$ and oracle $\mathcal{O}^{L_1}_{\mathsf{NIZK}}$ from the ZK challenger.
  2. Set $\mathsf{crs}^1_\Pi \leftarrow \mathsf{crs}_\Pi$; embed it into the experiment.
  3. Produce each honest $L_1$ proof in the signing oracle by querying $\mathcal{O}^{L_1}_{\mathsf{NIZK}}$ on the correct witness.
  4. Run $\mathcal{A}$ and output its bit. Any distinguishing advantage transfers to ZK.
```

**Game 2: Extractable $\mathsf{crs}^1_\Pi$.** Replace $\mathsf{crs}^1_\Pi$ by an extractable CRS $(\mathsf{crs}^1_\Pi, \tau^1, \zeta^1) \leftarrow_\$ \mathcal{E}(1^\lambda)$. All honest $L_1$ proofs remain simulated as before.

*Transition.* By SSE setup indistinguishability, the CRS distribution is preserved:

$$|\Pr[S_2] - \Pr[S_1]| = 0.$$

**Game 3: Open via extraction from $L_1$.** On a verifying forgery $\sigma^* = (c^*, \pi^*)$ for instance $x^* = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c^*, m^*)$, use the extractor for $\mathsf{crs}^1_\Pi$ to obtain $(r^*, \sigma^*_s, w^*)$. Set $y^* = f(w^*)$ and check

$$c^* \overset{?}{=} \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, y^*; r^*) \quad \wedge \quad \mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, y^*), \sigma^*_s) = 1.$$

Abort if extraction or recomputation fails.

*Transition.* If $\mathsf{Vfy}_\Pi(\mathsf{crs}^1_\Pi, x^*, \pi^*) = 1$ but extraction fails or recomputation is inconsistent, SSE is violated. There exists $\mathcal{B}^{(2)}_{\mathsf{sse}}$ such that

$$|\Pr[S_3] - \Pr[S_2]| \le \varepsilon^{(2)}_{\mathsf{sse}}(\lambda).$$

```
$\mathcal{B}^{(2)}_{\mathsf{sse}}(1^\lambda)$
  1. Receive an extractable CRS $\mathsf{crs}^1_\Pi$ (and oracle if provided) from the SSE challenger.
  2. Run $\mathcal{A}$ as in Game 2; answer honest $L_1$ proofs via the SSE-provided oracle.
  3. When $\mathcal{A}$ outputs a verifying $\pi^*$ on $x^*$, forward $(x^*, \pi^*)$ to the SSE challenger.
  4. If extraction fails or recomputation is inconsistent, this yields an SSE break.
```

**Advantage analysis.**

$$\Pr[S_0] \le \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{sse}}^{(2)}.$$

All terms are negligible, hence $\Delta'$ satisfies traceability. □

**Conclusion.** Any adversary that produces a verifying signature which the honest opener cannot link to a valid member necessarily yields a contradiction: either it distinguishes simulated from real $L_1$ proofs (breaking ZK), or it produces an $L_1$ proof that verifies but cannot be extracted/recomputed (breaking SSE). Both cases occur with negligible probability.

### D.4 Proof of Theorem 18 (Anonymity)

**Proof strategy.** We gradually transform the anonymity experiment so that the challenge ciphertext becomes independent of the hidden bit. Each hop either enforces openability of non-challenge signatures, replaces NIZKs by simulated ones, verifies openings via extraction, or replaces the challenge encryption by a dummy value. Any adversary that distinguishes along the way would contradict traceability, zero-knowledge, simulation-sound extractability, or mcINDCPA.

- **Game 0 (Original experiment).** Baseline anonymity experiment for $\Delta'$.
- **Game 1 (Openability check).** Ensure that every non-challenge signature can be opened with the honest opener. Otherwise, traceability is broken.
- **Game 2 (Simulated $L_1$).** Replace $\mathsf{crs}_{\Pi}^{1}$ by a simulated CRS and simulate all signature-side proofs. Indistinguishable by zero-knowledge.
- **Game 3 (Simulated $L_2$).** Replace $\mathsf{crs}_{\Pi}^{2}$ by a simulated CRS and simulate all opening proofs. Indistinguishable by zero-knowledge.
- **Game 4$_i$ (Extraction hybrids for $L_2$).** Gradually verify opening proofs by extraction from an extractable CRS. Any gap would contradict SSE.
- **Game 5 (Challenge ciphertext switch).** Replace the challenge encryption by an encryption of a fixed dummy value. Indistinguishable by mcINDCPA.

After Game 5, the challenge signature is independent of the hidden bit $b$, so $\mathcal{A}$'s success probability is exactly $1/2$. Thus any non-negligible advantage would break one of the underlying assumptions, and the scheme achieves anonymity.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against anonymity. We define the following games and write $S_i$ for the event that $\mathcal{A}$ outputs $b^* = b$ in Game $i$.

**Game 0: Original experiment.** This is the standard anonymity experiment for $\Delta'$. Let $S_0$ denote the event that $\mathcal{A}$ wins.

**Game 1: Openability check for non-challenge signatures.** Modify the signing oracle so that for every *non-challenge* query the challenger runs $\mathsf{Opn}_{\Delta}$ (with the honest opener key) on the returned signature and aborts if opening fails.

*Transition.* If some non-challenge signature cannot be opened, traceability is broken. There exists $\mathcal{B}_{\mathsf{trace}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \le \varepsilon_{\mathsf{trace}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{trace}}(1^{\lambda})}$

1. Interact with the traceability challenger (environment as in $\Delta'$).
2. Run Game 0 with $\mathcal{A}$. For every non-challenge signing query, invoke the honest opener on the produced signature.
3. If any such signature fails to open, output it as a witness of traceability violation.

**Game 2: Simulate $\mathsf{crs}^1_{\Pi}$.** Generate $(\mathsf{crs}^1_{\Pi}, \tau^1) \leftarrow_{\$} \mathsf{SIM}_1(1^{\lambda})$ and *simulate* all signature-side NIZK proofs $\pi \in L_1$ (every honest proof is obtained from the ZK oracle on the correct witness; whether it simulates or proves is hidden).

*Transition.* By ZK there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ with

$$|\Pr[S_2] - \Pr[S_1]| \leq \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(1)}_{\mathsf{zk}}(1^{\lambda})}$

1. Receive CRS $\mathsf{crs}_{\Pi}$ and access to $\mathcal{O}^{L_1}_{\mathsf{NIZK}}$ from the ZK challenger.
2. Set $\mathsf{crs}^1_{\Pi} \leftarrow \mathsf{crs}_{\Pi}$.
3. Whenever Game 2 needs an honest $L_1$ proof on $x = (\mathsf{pk}_{\Sigma}, \mathsf{pk}_{\Theta}, c, m)$ with witness $(r, \sigma_s, w)$, query $\pi \leftarrow \mathcal{O}^{L_1}_{\mathsf{NIZK}}(\mathsf{crs}^1_{\Pi}, x; (r, \sigma_s, w))$.
4. Run $\mathcal{A}$ and output its bit. Any distinguishing advantage transfers to ZK.

**Game 3: Simulate $\mathsf{crs}^2_{\Pi}$.** Generate $(\mathsf{crs}^2_{\Pi}, \tau^2) \leftarrow_{\$} \mathsf{SIM}_1(1^{\lambda})$ and *simulate* all opening proofs $\pi_o \in L_2$.

*Transition.* By ZK there exists $\mathcal{B}^{(2)}_{\mathsf{zk}}$ with

$$|\Pr[S_3] - \Pr[S_2]| \leq \varepsilon^{(2)}_{\mathsf{zk}}(\lambda).$$

$\underline{\mathcal{B}^{(2)}_{\mathsf{zk}}(1^{\lambda})}$

1. Receive CRS $\mathsf{crs}_{\Pi}$ and access to $\mathcal{O}^{L_2}_{\mathsf{NIZK}}$ from the ZK challenger.
2. Set $\mathsf{crs}^2_{\Pi} \leftarrow \mathsf{crs}_{\Pi}$.
3. Whenever Game 3 needs an honest $L_2$ proof on $x_o = (\mathsf{pk}_{\Sigma}, \mathsf{pk}_{\Theta}, c, m, y)$ with witness $\mathsf{sk}_{\Theta}$, query $\pi_o \leftarrow \mathcal{O}^{L_2}_{\mathsf{NIZK}}(\mathsf{crs}^2_{\Pi}, x_o; \mathsf{sk}_{\Theta})$.
4. Run $\mathcal{A}$ and output its bit.

**Game $4_i$ hybrids for $0 \leq i \leq q_o$.** For the first $i$ (non-challenge) opening queries, verify by *extraction* under an extractable CRS for $\mathsf{crs}^2_{\Pi}$: for each accepting $\pi_o \in L_2$ on $x_o = (\mathsf{pk}_{\Sigma}, \mathsf{pk}_{\Theta}, c, m, y)$ extract $\mathsf{sk}_{\Theta}$ and check $\mathsf{KVfy}_{\Theta}(\mathsf{sk}_{\Theta}, \mathsf{pk}_{\Theta}) = 1$ and $\mathsf{Dec}_{\Theta}(\mathsf{sk}_{\Theta}, c) = y$. All remaining queries are checked by $\mathsf{Vfy}_{\Pi}$. Game $4_0$=Game 3 and Game $4_{q_o}$=Game 4.

*Transition.* If the $i$-th hybrid distinguishes, then a proof verifies but extraction or recomputation fails, contradicting SSE. Hence there exists $\mathcal{B}^{(2,i)}_{\mathsf{sse}}$ such that

$$|\Pr[S^{(i)}_4] - \Pr[S^{(i-1)}_4]| \leq \varepsilon^{(2)}_{\mathsf{sse}}(\lambda).$$

$\underline{\mathcal{B}^{(2,i)}_{\mathsf{sse}}(1^{\lambda})}$

1. Receive an *extractable* CRS $\hat{\mathsf{crs}}_{\Pi}$ for $L_2$ and a simulation oracle $\mathsf{SIM}^{L_2}_2$ from the SSE challenger.
2. Set $\mathsf{crs}^2_{\Pi} \leftarrow \hat{\mathsf{crs}}_{\Pi}$. Answer all honest $L_2$ proofs using $\mathsf{SIM}^{L_2}_2$.
3. On the $i$-th opening query by $\mathcal{A}$, when it outputs a verifying $(x^*_o, \pi^*_o)$, forward $(x^*_o, \pi^*_o)$ to the SSE challenger to test extractability/consistency.
4. Continue the interaction and output $\mathcal{A}$'s bit.

**Game 5: Replace the challenge encryption.** In the *challenge* signing oracle, replace $c^*$ by an encryption of a fixed dummy value, $c^* \leftarrow \mathsf{Enc}_{\Theta}(\mathsf{pk}_{\Theta}, 0)$. All NIZKs remain simulated as before.

*Transition.* This hop is indistinguishable by mcINDCPA of $\Theta$. There exists $\mathcal{B}_{\mathsf{mcindcpa}}$ with

$$|\Pr[S_5] - \Pr[S^{(q_o)}_4]| \leq \varepsilon_{\mathsf{mcindcpa}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{mcindcpa}}(1^\lambda)}$
1. Receive $\mathsf{pp}_\Theta$ and $\mathsf{pk}_\Theta$ from the mcINDCPA challenger (with LR oracle).
2. In the challenge signing query $(\mathsf{upk}^{(0)}, \mathsf{upk}^{(1)}, m^*)$, recover the corresponding identities $y_0 = f(w_0)$, $y_1 = f(w_1)$. Submit $(y_0, y_1)$ to the LR oracle to obtain a challenge ciphertext $c^*$.
3. Embed $c^*$ into the challenge signature and continue the Game 5 interaction with $\mathcal{A}$ (all NIZKs are simulated).
4. When $\mathcal{A}$ outputs $b^*$, forward $b^*$ to the mcINDCPA challenger.
```

**Advantage analysis.** Let $\varepsilon(\lambda) = \Pr[S_0] - \frac{1}{2}$. Then

$$|\varepsilon(\lambda)| \;\leq\; \varepsilon_{\mathsf{trace}} + \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{zk}}^{(2)} + q_o \cdot \varepsilon_{\mathsf{sse}}^{(2)} + \varepsilon_{\mathsf{mcindcpa}},$$

which is negligible. Therefore $\Delta'$ satisfies anonymity. □

**Conclusion.** At this point, the challenge signature is independent of the hidden bit $b$: all NIZK proofs are simulated (thus independent of witnesses), all non-challenge openings are checked by extraction/consistency (ruling out malleability attacks), and the challenge ciphertext no longer encodes $f(w_b)$ (by the mcINDCPA hop). Hence $\Pr[S_5] = \frac{1}{2}$ up to negligible terms.

### D.5 Proof of Theorem 19 (Non-Claimability)

**Proof strategy.** We transform the non-claimability experiment step by step. Each hop either replaces the CRS by a simulated or extractable one, or enforces consistency checks by extraction and decryption. Any adversary that survives these steps would contradict zero-knowledge, simulation-sound extractability, or decryption soundness.

- **Game 0 (Original experiment).** Baseline non-claimability experiment: all logged signatures are honestly generated for the fixed honest member identity.
- **Game 1 (Simulated $L_2$).** Replace $\mathsf{crs}_\Pi^2$ by a simulated CRS. Indistinguishable by zero-knowledge.
- **Game 2 (Extractable $L_2$).** Switch to an extractable CRS for $\mathsf{crs}_\Pi^2$. Distribution preserved by SSE.
- **Game 3 (Extraction from $L_2$).** Extract the decryption key from any accepting $L_2$ proof and check consistency. Otherwise, SSE is broken.
- **Game 4 (Decryption soundness).** Check that extracted keys correctly decrypt honestly generated ciphertexts. Otherwise, Decryption Soundness is broken.

After Game 4, no adversary can output a transcript from the log that opens to a different identity: all openings are consistent and bound to the fixed honest member. Thus any successful adversary would contradict ZK, SSE, or DS, and the scheme satisfies non-claimability.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against non-claimability (Fig. 8). We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Real experiment.** This is $\mathbf{Exp}_{\mathcal{A},\Delta'}^{\mathsf{NonClaimability}}(\lambda)$. By definition of $\mathcal{O}_{\mathsf{Sgn}_\Delta}$, every logged signature is honestly generated as

$$\sigma_m = (c, \pi) \quad \text{with} \quad c = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta, y; r)$$

for the fixed honest identity $y = f(w)$ and coins $r$ (we record $r$ internally alongside the log entry). Let $S_0$ be the event that $\mathcal{A}$ outputs $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, \mathsf{upk}^*, O^*)$ with $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*) \in \mathcal{Q}$ but $\mathsf{upk}^* \neq \mathsf{upk}$ and $\mathsf{Jdg}_\Delta(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, \mathsf{upk}^*, O^*) = 1$.

**Game 1: Simulate $\mathsf{crs}_\Pi^2$.** Generate $(\mathsf{crs}_\Pi^2, \tau^2) \leftarrow_\$ \mathsf{SIM}_1(1^\lambda)$ and use it for all $L_2$ verifications. (There are no honest $L_2$ proofs generated by the challenger in this experiment.)

*Transition.* By ZK there exists $\mathcal{B}_{\mathsf{zk}}^{(1)}$ such that

$$\left|\Pr[S_1] - \Pr[S_0]\right| \leq \varepsilon_{\mathsf{zk}}^{(1)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{zk}}^{(1)}(1^\lambda)}$

1. Receive $\mathsf{crs}_\Pi$ from the ZK challenger.
2. Set $\mathsf{crs}_\Pi^2 \leftarrow \mathsf{crs}_\Pi$; keep the rest of the environment as in Game 0, but ZK-proofs are relayed to the oracle provided.
3. Run $\mathcal{A}$ to completion and output its bit. Any non-negligible gap distinguishes the hops.

**Game 2: Extractable $\mathsf{crs}_\Pi^2$.** Replace $\mathsf{crs}_\Pi^2$ by an extractable setup $(\mathsf{crs}_\Pi^2, \tau^2, \zeta^2) \leftarrow_\$ \mathcal{E}(1^\lambda)$. No other change.

*Transition.* By SSE (setup indistinguishability),

$$\left|\Pr[S_2] - \Pr[S_1]\right| = 0.$$

**Game 3: Abort on $L_2$ extraction failure / wrong witness.** On an accepting output with $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*) \in \mathcal{Q}$ and $\mathsf{upk}^* \neq \mathsf{upk}$, parse $\sigma_m^* = (c^*, \pi^*)$ and $O^* = \pi_o^*$, set $x_o^* = (\mathsf{pk}_\Sigma^*, \mathsf{pk}_\Theta^*, c^*, m^*, y^*)$ with $y^* = \mathsf{upk}^*$, extract $\mathsf{sk}_\Theta^*$ from $\pi_o^*$, and check

$$\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*) = 1 \qquad \wedge \qquad \mathsf{Dec}_\Theta(\mathsf{sk}_\Theta^*, c^*) = y^*.$$

Abort if extraction fails or a recomputation check fails.

*Transition.* If $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi^2, x_o^*, \pi_o^*) = 1$ but extraction or recomputation fails, SSE is violated. There exists $\mathcal{B}_{\mathsf{sse}}^{(3)}$ with

$$\left|\Pr[S_3] - \Pr[S_2]\right| \leq \varepsilon_{\mathsf{sse}}^{(3)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}^{(3)}(1^\lambda)}$

1. Receive an extractable CRS $\hat{\mathsf{crs}}_\Pi$ for $L_2$ (and a simulation oracle $\mathsf{SIM}_2^{L_2}$, if provided) from the SSE challenger.
2. Set $\mathsf{crs}_\Pi^2 \leftarrow \hat{\mathsf{crs}}_\Pi$; run $\mathcal{A}$ in the Game 2 environment (no honest $L_2$ proofs are needed).
3. When $\mathcal{A}$ outputs an accepting $(x_o^*, \pi_o^*)$, forward it to the SSE challenger.

**Game 4: Abort on Decryption-Soundness violation.** Because $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*) \in \mathcal{Q}$, there exist coins $r^*$ (recorded in the log) such that $c^* = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta^*, y; r^*)$ for the fixed honest $y = f(w)$. If additionally $\mathsf{KVfy}_\Theta(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*) = 1$ but $\mathsf{Dec}_\Theta(\mathsf{sk}_\Theta^*, c^*) \neq y$, abort.

*Transition.* This breaks Decryption Soundness. There exists $\mathcal{B}_{\mathsf{ds}}$ such that

$$\left|\Pr[S_4] - \Pr[S_3]\right| \leq \varepsilon_{\mathsf{ds}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{ds}}(1^\lambda)}$

1. Receive $\mathsf{pp}_\Theta$ from the DecSound challenger.
2. Run $\mathcal{A}$ in the Game 3 environment to obtain $(\mathsf{ipk}^*, \mathsf{opk}^*, m^*, \sigma_m^*, \mathsf{upk}^*, O^*)$.
3. From the log, recover the coins $r^*$ and $\mathsf{pk}_\Theta^*$ with $c^* = \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta^*, y; r^*)$ for the fixed honest $y$.
4. From $O^* = \pi_o^*$ extract $\mathsf{sk}_\Theta^*$ consistent with $\mathsf{pk}_\Theta^*$, and output $(\mathsf{sk}_\Theta^*, \mathsf{pk}_\Theta^*, y, r^*)$ to the DecSound challenger.

**Advantage analysis.** By a union bound,

$$\Pr[S_0] \;\le\; \varepsilon_{\mathsf{zk}}^{(1)} \;+\; \varepsilon_{\mathsf{sse}}^{(3)} \;+\; \varepsilon_{\mathsf{ds}},$$

which is negligible. □

**Conclusion.** Any adversary that reassigns an honestly logged transcript to a different member must either (i) produce an $L_2$ proof that verifies but fails extraction or recomputation (contradicting SSE), or (ii) provide a valid key for which decryption of an honestly formed ciphertext yields a value different from the planted $y$ (contradicting Decryption Soundness). The initial CRS swap for $L_2$ is hidden by ZK. Each event occurs with only negligible probability, hence $\Delta'$ satisfies non-claimability.

### D.6 Proof of Theorem 20 (Certificate-Unforgeability)

**Proof strategy.** We transform the certificate-unforgeability experiment step by step. Each hop either replaces the CRS by a simulated or extractable one, or extracts witnesses from the adversary's forgery. In the end, any successful forgery must yield either a break of SSE or a break of DSIG unforgeability.

- **Game 0 (Original experiment).** Baseline certificate-unforgeability experiment: the adversary must output a valid, non-replayed group signature although it never obtained a join certificate.
- **Game 1 (Simulated CRS).** Replace all NIZK CRS used inside oracles by simulated CRS and produce proofs via the ZK simulator. Indistinguishable by zero-knowledge.
- **Game 2 (Extractable CRS).** Switch to an extractable CRS for the proofs that occur in the adversary's final forgery. Distribution preserved (distance 0).
- **Game 3 (Extraction).** Apply the SSE extractor to the forged proof(s). If extraction or recomputation fails, then SSE is broken.
- **Game 4 (DSIG reduction).** If the extracted certificate $\sigma_s^*$ is new, then $(\mathsf{pk}_\Sigma, \sigma_s^*)$ is a valid DSIG forgery. Otherwise the forgery contradicts DSIG unforgeability of the signature component.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against certificate-unforgeability. We define a sequence of games; $S_i$ denotes the success event in Game $i$.

**Game 0: Original experiment.** As in Fig. 9a. Let $S_0$ be the event that $\mathcal{A}$ outputs a valid signature $\sigma_m^*$ on message $m^*$ with $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}^*, m^*, \sigma_m^*) = 1$, not a replay of any element in $\mathcal{Q}_\sigma$.

**Game 1: Simulated CRS.** Generate $(\mathsf{crs}_\Pi, \tau) \leftarrow_\$ \mathsf{SIM}(1^\lambda)$ for all CRS used in oracle proofs. Answer all oracle queries by the simulator.

*Transition.* By zero-knowledge there exists $\mathcal{B}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \le \varepsilon_{\mathsf{zk}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{zk}}(1^\lambda)}$    1. Receive CRS $\mathsf{crs}_\Pi$ and oracle access $\mathcal{O}_{\mathsf{NIZK}}$ from the ZK challenger.
    2. Embed $\mathsf{crs}_\Pi$ into $\mathsf{pp}_\Delta$ and run $\mathcal{A}$.
    3. Whenever an oracle must produce a proof on instance $x$ with witness $w$, query $\pi \leftarrow \mathcal{O}_{\mathsf{NIZK}}(\mathsf{crs}_\Pi, x; w)$
       and return $\pi$.
    4. Output whatever $\mathcal{A}$ outputs. Any distinguishing advantage transfers to the ZK challenger.
```

**Game 2: Extractable CRS.** Replace $\mathsf{crs}_\Pi$ by an extractable setup $(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_\$ \mathcal{E}(1^\lambda)$. Honest oracle proofs remain simulated as in Game 1.

*Transition.* By SSE setup indistinguishability,

$$|\Pr[S_2] - \Pr[S_1]| = 0.$$

$\underline{\mathcal{B}_{\mathsf{setup}}(1^\lambda)}$ 1. Receive $(\mathsf{crs}_\Pi, \tau, \zeta)$ from the SSE challenger.
2. Use $\mathsf{crs}_\Pi$ for the adversary's environment; answer oracle proofs via simulation as in Game 1.
3. Relay $\mathcal{A}$ unchanged. Distribution remains identical.

**Game 3: Extraction.** On an accepting forgery $\sigma_m^*$ for message $m^*$, apply the extractor to obtain witnesses $(r^*, \sigma_s^*, w^*)$. Abort if extraction or recomputation fails.

*Transition.* Otherwise SSE is broken. There exists $\mathcal{B}_{\mathsf{sse}}$ with

$$|\Pr[S_3] - \Pr[S_2]| \leq \varepsilon_{\mathsf{sse}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}(1^\lambda)}$ 1. Fix $\mathsf{crs}_\Pi$ to an extractable CRS as in Game 2.
2. Run $\mathcal{A}$; on any accepting forgery $(x^*, \pi^*)$, forward to the SSE challenger.
3. If extraction fails, $\mathcal{B}_{\mathsf{sse}}$ wins.

**Game 4: Reduction to $\Sigma$ unforgeability.** If the extracted certificate $\sigma_s^*$ was never produced by the join oracle, then $(\mathsf{pk}_\Sigma, \sigma_s^*)$ constitutes a valid DSIG forgery.

*Transition.* There exists $\mathcal{B}_{\mathsf{dsigunf}}$ with

$$|\Pr[S_4] - \Pr[S_3]| \leq \varepsilon_{\mathsf{dsigunf}}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{dsigunf}}(1^\lambda)}$ 1. Receive verification key $\mathsf{pk}_\Sigma$ and signing oracle access from the DSIG challenger.
2. Embed $\mathsf{pk}_\Sigma$ into the GS parameters and answer join/sign requests via the oracle.
3. Run $\mathcal{A}$; on a successful forgery $\sigma_m^*$ with extracted $\sigma_s^*$ never joined, output $(x^*, \sigma_s^*)$ as a DSIG forgery.

**Advantage analysis.** By a union bound we obtain

$$\Pr[S_0] \leq \varepsilon_{\mathsf{zk}} + \varepsilon_{\mathsf{sse}} + \varepsilon_{\mathsf{dsigunf}}.$$

All terms are negligible, hence $\Delta'$ satisfies certificate-unforgeability. □

**Conclusion.** Any adversary that outputs a valid, non-replayed group signature without an issued certificate must either break SSE extractability or forge a DSIG signature. Both occur only with negligible probability.

### D.7 Proof of Theorem 21 (Key-Unforgeability

**Proof strategy.** We run six games. We first switch to a simulated, then to an extractable CRS for $L_1$. We enforce witness consistency by extraction. A successful forgery either yields a $\Sigma$ forgery (never-joined identity) or lets us invert $f$ on a registered identity. The latter is handled via an OWF reduction with an index guess over member generations.

– **Game 0 (Original).** Baseline experiment.
– **Game 1 (Simulated $L_1$).** Replace $\mathsf{crs}_\Pi^1$ by a simulated CRS; answer all honest $L_1$ proofs via the ZK simulator.

- **Game 2 (Extractable $L_1$).** Switch to an SSE CRS for $L_1$ that supports both simulation and extraction.
- **Game 3 (Extraction check).** On any accepting forgery, extract $(r^*, \sigma_s^*, w^*)$ and check ciphertext/signature consistency; else contradict SSE.
- **Game 4 ($\Sigma$ reduction).** If the extracted identity was never joined, output a $\Sigma$ forgery.
- **Game 5 (OWF reduction).** Guess one registered-identity index and embed an OWF challenge into that $\mathsf{KgM}_\Delta$ answer; if the forgery hits it, output the extracted preimage.

*Proof.* Let $\mathcal{A}$ be PPT. Let $S_i$ denote adversarial success in Game $i$.

**Game 0: Original.** As in Fig. 9b. Let $S_0$ be the event that $\mathcal{A}$ outputs $\sigma_m^* = (c^*, \pi^*)$ on $m^*$ with $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}^*, m^*, \sigma_m^*) = 1$ and $(\sigma_m^*, m^*, \mathsf{opk}^*, \mathsf{ipk}) \notin \mathcal{Q}_\sigma$.

**Game 1: Simulated $\mathsf{crs}_\Pi^1$.** Sample $(\mathsf{crs}_\Pi^1, \tau^1) \leftarrow \mathsf{SIM}_1(1^\lambda)$. Answer every honest $L_1$ proof in $\mathcal{O}_{\mathsf{Sgn}_\Delta}$ via the ZK simulator; all other algorithms as specified.

*Transition (ZK).* There exists $\mathcal{B}_{\mathsf{zk}}^{(1)}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \le \varepsilon_{\mathsf{zk}}^{(1)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{zk}}^{(1)}(1^\lambda)}$

1. Receive $(\mathsf{crs}_\Pi^1, \mathcal{O}_{\mathsf{Sim}}^{L_1})$ from the ZK challenger.
2. Run $\mathsf{PG}_\Delta$ to get $\mathsf{pp}_\Delta$, then $\mathsf{KgI}_\Delta$ to get $(\mathsf{isk}, \mathsf{ipk})$; give $\mathsf{ipk}$ to $\mathcal{A}$.
3. Provide real oracles, except: when $\mathcal{O}_{\mathsf{Sgn}_\Delta}$ must prove $x = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta, c, m)$, set $\pi \leftarrow \mathcal{O}_{\mathsf{Sim}}^{L_1}(\mathsf{crs}_\Pi^1, x)$ and return $\sigma_m = (c, \pi)$.
4. Output $\mathcal{A}$'s bit to the ZK challenger.

**Game 2: Extractable $\mathsf{crs}_\Pi^1$.** Replace $\mathsf{crs}_\Pi^1$ by SSE CRS $(\mathsf{crs}_\Pi^1, \tau^1, \zeta^1) \leftarrow \mathcal{E}(1^\lambda)$. Continue simulating all honest $L_1$ proofs; adversarial proofs become extractable.

*Transition (SSE-setup).*

$$|\Pr[S_2] - \Pr[S_1]| = 0.$$

$\underline{\mathcal{B}_{\mathsf{sse\text{-}set}}^{(2)}(1^\lambda)}$

1. Receive either $(\mathsf{crs}_\Pi^1, \tau^1)$ or $(\mathsf{crs}_\Pi^1, \tau^1, \zeta^1)$ from the SSE challenger.
2. Run Game 1 with the provided $\mathsf{crs}_\Pi^1$; simulate all honest $L_1$ proofs.
3. Output $\mathcal{A}$'s bit; any gap distinguishes the two setups.

**Game 3: Extraction and consistency.** On a successful output $\sigma_m^* = (c^*, \pi^*)$ for $m^*$, set $x^* = (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta^*, c^*, m^*)$. Extract $(r^*, \sigma_s^*, w^*) \leftarrow \mathsf{Ext}(\zeta^1, \pi^*, x^*)$; set $y^\dagger \leftarrow f(w^*)$, and check

$$c^* \overset{?}{=} \mathsf{Enc}_\Theta(\mathsf{pk}_\Theta^*, y^\dagger; r^*), \qquad \mathsf{Vfy}_\Sigma\big(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}_\Theta^*, y^\dagger), \sigma_s^*\big) = 1.$$

Abort if any check fails.

*Transition (SSE-extraction).* There exists $\mathcal{B}_{\mathsf{sse}}^{(3)}$ with

$$|\Pr[S_3] - \Pr[S_2]| \le \varepsilon_{\mathsf{sse}}^{(2)}(\lambda).$$

$\underline{\mathcal{B}_{\mathsf{sse}}^{(3)}(1^\lambda)}$

1. Obtain $(\mathsf{crs}_\Pi^1, \tau^1, \zeta^1)$ from the SSE challenger; run $\mathcal{A}$ as in Game 2.
2. On $\sigma_m^*$, form $x^*$ and compute $(r^*, \sigma_s^*, w^*) \leftarrow \mathsf{Ext}(\zeta^1, \pi^*, x^*)$.
3. If extraction or a recomputation check fails, output 1 to the SSE challenger; else output 0.

**Game 4: $\Sigma$ reduction.** Let $E_{\mathrm{noj}}$ be the event that the extracted $y^{\dagger}$ was never joined for $\mathsf{pk}_{\Theta}^{*}$ (and the output is not a replay). On $E_{\mathrm{noj}}$, $\sigma_s^*$ is a valid $\Sigma$ forgery on $(\mathsf{pk}_{\Sigma}, \mathsf{pk}_{\Theta}^{*}, y^{\dagger})$.

*Transition (DSIG-EUF).* There exists $\mathcal{B}_{\mathsf{dsig}}$ such that

$$\Pr[S_4 \wedge E_{\mathrm{noj}}] \leq \varepsilon_{\mathsf{dsigunf}}(\lambda), \qquad |\Pr[S_4] - \Pr[S_3]| \leq \varepsilon_{\mathsf{dsigunf}}.$$

```
$\underline{\mathcal{B}_{\mathsf{dsig}}(1^{\lambda})}$
  1. Receive $\mathsf{pk}_{\Sigma}$ and signing oracle $\mathcal{O}_{\Sigma}$ from the DSIG challenger; embed $\mathsf{pk}_{\Sigma}$ in $\mathsf{pp}_{\Delta}$.
  2. Answer: $\mathcal{O}_{\mathsf{KgM}_{\Delta}}$ honestly; store $(w, y)$. $\mathcal{O}_{\mathsf{Join}_{\Delta}}(\mathsf{opk}, y)$ via $\sigma_s \leftarrow \mathcal{O}_{\Sigma}((\mathsf{pk}_{\Sigma}, \mathsf{opk}, y))$. $\mathcal{O}_{\mathsf{Sgn}_{\Delta}}$ by simulating $L_1$ proofs; encryption per scheme.
  3. On $\sigma_m^*$, extract $(r^*, \sigma_s^*, w^*)$; set $y^{\dagger} = f(w^*)$. If $E_{\mathrm{noj}}$ holds and not a replay, output $((\mathsf{pk}_{\Sigma}, \mathsf{opk}^*, y^{\dagger}), \sigma_s^*)$.
```

**Game 5: OWF reduction.** Let $q_{\mathsf{kgen}}$ bound the number of $\mathcal{O}_{\mathsf{KgM}_{\Delta}}$ queries. The OWF reduction $\mathcal{B}_{\mathsf{owf}}$ is given a challenge $y^* = f(w^*)$. Pick $i^* \leftarrow_{\$} \{1, \ldots, q_{\mathsf{kgen}}\}$. Answer the $j$-th $\mathcal{O}_{\mathsf{KgM}_{\Delta}}$ by: if $j = i^*$, return $\mathsf{upk} \leftarrow y^*$ and record $(\perp, y^*)$; else sample $(w, y)$ honestly and return $y$. Answer $\mathcal{O}_{\mathsf{Join}_{\Delta}}$ and $\mathcal{O}_{\mathsf{Sgn}_{\Delta}}$ as in Game 3 (joins real or via $\mathcal{B}_{\mathsf{dsig}}$; $L_1$ proofs simulated). Run $\mathcal{A}$. On its successful output, extract $(r^*, \sigma_s^*, w^{\dagger})$ and set $y^{\dagger} = f(w^{\dagger})$. If $y^{\dagger} = y^*$, output $w^{\dagger}$ to invert $f$.

*Analysis (OWF).* Let $E_{\mathrm{hit}}$ be the event that the forgery's extracted identity $y^{\dagger}$ equals the $i^*$-th registered identity. Then $\mathcal{B}_{\mathsf{owf}}$ succeeds with probability $\Pr[E_{\mathrm{hit}}]/q_{\mathsf{kgen}}$, hence

$$\Pr[E_{\mathrm{hit}}] \leq q_{\mathsf{kgen}} \cdot \varepsilon_{\mathsf{owf}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{owf}}(1^{\lambda})}$
  1. Receive $y^* = f(w^*)$; pick $i^*$; embed $y^*$ into the $i^*$-th $\mathsf{KgM}_{\Delta}$ response.
  2. Run $\mathcal{A}$ with Game 3 oracles ($L_1$ proofs simulated); on success extract $w^{\dagger}$.
  3. If $f(w^{\dagger}) = y^*$, return $w^{\dagger}$; else return $\perp$.
```

**Advantage analysis.** Split success in Game 4 into disjoint cases $E_{\mathrm{noj}}$ and $\neg E_{\mathrm{noj}}$. The former is bounded by $\varepsilon_{\mathsf{dsigunf}}$; the latter implies a hit on some registered identity and contributes at most $q_{\mathsf{kgen}}\varepsilon_{\mathsf{owf}}$. Combining all transitions,

$$\Pr[S_0] \;\leq\; \varepsilon_{\mathsf{zk}}^{(1)} + \varepsilon_{\mathsf{sse}}^{(2)} + \varepsilon_{\mathsf{dsigunf}} + q_{\mathsf{kgen}} \cdot \varepsilon_{\mathsf{owf}}.$$

All terms are negligible for polynomially bounded queries. Hence $\Delta'$ is key-unforgeable. □

**Conclusion.** Either the forgery targets a never-joined identity, yielding a $\Sigma$ forgery, or it hits a registered identity, yielding an OWF preimage. Both occur with negligible probability.

### D.8 Proof of Theorem 22 (Opening-Unforgeability)

**Proof strategy.** We transform the opening-unforgeability experiment in several steps. Each hop either replaces the CRS by a simulated or extractable one, or enforces consistency checks by extraction and decryption. Any adversary that still succeeds must necessarily forge a DSIG signature or break one of the underlying assumptions.

- **Game 0 (Original experiment).** Baseline: adversary outputs a valid opening for a member/opener pair never recorded in the transcript.

- **Game 1–2 (Simulated CRS).** Replace the CRS for $L_1$ and $L_2$ by simulated ones. Indistinguishable by zero-knowledge.
- **Game 3–4 (Extractable CRS).** Switch both CRSs to extractable versions. Distribution preserved by SSE setup indistinguishability.
- **Game 5 (Extraction from $L_1$).** Extract from the forged $L_1$ proof and check ciphertext/signature consistency. Otherwise, SSE is broken.
- **Game 6 (DSIG unforgeability).** If the extracted identity was never joined, then the auxiliary DSIG signature is a fresh forgery.
- **Game 7 (Extraction from $L_2$).** Extract the decryption key from the $L_2$ proof and check key validity. Otherwise, SSE is broken.
- **Game 8 (Decryption Soundness).** Check that extracted keys correctly decrypt the honestly generated ciphertext. Otherwise, DS is broken.

After Game 8, any successful opening forgery is impossible: all proofs are consistent, ciphertexts decrypt correctly, and no unjoined identity can be certified. Hence any adversary breaking opening unforgeability would contradict ZK, SSE, DSIG unforgeability, or DS.

*Proof.* Let $\mathcal{A}$ be a PPT adversary against opening unforgeability, acting as a malicious opener and interacting with an honest issuer. We define a sequence of games; $S_i$ denotes the event that $\mathcal{A}$ wins in Game $i$.

**Game 0: Original Experiment.** This is the standard experiment from Fig. 10. Let $S_0$ be the event that $\mathcal{A}$ outputs $(\mathsf{opk}^*, \mathsf{upk}^*, m^*, \sigma^*_m, O^*)$ with $(\mathsf{opk}^*, \mathsf{upk}^*) \notin \mathcal{Q}$ such that $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}^*, m^*, \sigma^*_m, \mathsf{upk}^*, O^*) = 1$.

**Game 1: Simulated CRS $\mathsf{crs}^1_\Pi$.** We now replace the CRS for $L_1$ by a simulated CRS: $(\mathsf{crs}^1_\Pi, \tau^1) \leftarrow \mathsf{SIM}_1(1^\lambda)$, and generate all honest $L_1$ proofs using the simulator. All other components remain unchanged.

*Transition.* If $\mathcal{A}$ distinguishes this change, we obtain a distinguisher for zero-knowledge of $L_1$: there exists $\mathcal{B}^{(1)}_{\mathsf{zk}}$ such that

$$|\Pr[S_1] - \Pr[S_0]| \leq \varepsilon^{(1)}_{\mathsf{zk}}(\lambda).$$

```
$\mathcal{B}^{(1)}_{\mathsf{zk}}(1^\lambda)$
    1. Receive $\mathsf{crs}_\Pi$ and oracle $\mathsf{SIM}_2^{L_1}$ from the ZK challenger; set $\mathsf{crs}^1_\Pi \leftarrow \mathsf{crs}_\Pi$.
    2. Answer all honest $L_1$ proofs by querying $\mathsf{SIM}_2^{L_1}$.
    3. Run $\mathcal{A}$ in this environment and forward its output bit.
```

**Game 2: Simulated CRS $\mathsf{crs}^2_\Pi$.** Analogously, replace the CRS for $L_2$ by a simulated one, using $(\mathsf{crs}^2_\Pi, \tau^2) \leftarrow \mathsf{SIM}_1(1^\lambda)$. All honest $L_2$ proofs are generated via simulation.

*Transition.* If $\mathcal{A}$ distinguishes, we break ZK for $L_2$. There exists $\mathcal{B}^{(2)}_{\mathsf{zk}}$ such that

$$|\Pr[S_2] - \Pr[S_1]| \leq \varepsilon^{(2)}_{\mathsf{zk}}(\lambda).$$

```
$\mathcal{B}^{(2)}_{\mathsf{zk}}(1^\lambda)$
    1. Receive $\mathsf{crs}_\Pi$ and oracle $\mathsf{SIM}_2^{L_2}$ from the ZK challenger; set $\mathsf{crs}^2_\Pi \leftarrow \mathsf{crs}_\Pi$.
    2. Answer all honest $L_2$ proofs by querying $\mathsf{SIM}_2^{L_2}$.
    3. Run $\mathcal{A}$ and forward its output bit.
```

**Game 3: Extractable $\mathsf{crs}^1_\Pi$.** Switch to an extractable CRS for $L_1$: $(\mathsf{crs}^1_\Pi, \tau^1, \zeta^1) \leftarrow \mathcal{E}(1^\lambda)$. All honest $L_1$ proofs are still generated via simulation.

*Transition.* By SSE setup indistinguishability,

$$|\Pr[S_3] - \Pr[S_2]| = 0.$$

**Game 4: Extractable $\mathsf{crs}^2_\Pi$.** Analogously, switch to an extractable CRS for $L_2$: $(\mathsf{crs}^2_\Pi, \tau^2, \zeta^2) \leftarrow \mathcal{E}(1^\lambda)$.

*Transition.* Again by SSE indistinguishability,

$$|\Pr[S_4] - \Pr[S_3]| = 0.$$

**Game 5: Extraction from $\pi^*$ (L1).** For the forgery $\sigma^*_m = (c^*, \pi^*)$, form $x^* = (\mathsf{pk}_\Sigma, \mathsf{pk}^*_\Theta, c^*, m^*)$ and extract $(r^*, \sigma^*_s, w^*)$ from $\pi^*$. Check consistency:

$$c^* \overset{?}{=} \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, f(w^*); r^*) \quad \wedge \quad \mathsf{Vfy}_\Sigma(\mathsf{pk}_\Sigma, (\mathsf{pk}_\Sigma, \mathsf{pk}^*_\Theta, f(w^*)), \sigma^*_s) = 1.$$

Abort on failure.

*Transition.* If $\pi^*$ verifies but extraction fails, SSE is broken. There exists $\mathcal{B}^{(3)}_{\mathsf{sse}}$ such that

$$|\Pr[S_5] - \Pr[S_4]| \leq \varepsilon^{(3)}_{\mathsf{sse}}(\lambda).$$

```
$\mathcal{B}^{(3)}_{\mathsf{sse}}(1^\lambda)$
    1. Receive $\mathsf{crs}^1_\Pi$ with extractor from the SSE challenger.
    2. Run $\mathcal{A}$ in the Game 4 environment.
    3. On an accepting $(x^*, \pi^*)$, forward it to the SSE challenger.
```

**Game 6: DSIG unforgeability.** If the extracted $y^* = f(w^*)$ corresponds to a member that never joined (i.e. $(\mathsf{opk}^*, \mathsf{upk}^*) \notin \mathcal{Q}$), then $\sigma^*_s$ is a fresh signature under $\mathsf{pk}_\Sigma$.

*Transition.* Otherwise DSIG unforgeability is violated. There exists $\mathcal{B}_{\mathsf{dsig}}$ with

$$|\Pr[S_6] - \Pr[S_5]| \leq \varepsilon_{\mathsf{dsigunf}}(\lambda).$$

```
$\mathcal{B}_{\mathsf{dsig}}(1^\lambda)$
    1. Receive $\mathsf{pk}_\Sigma$ and signing oracle from the DSIG challenger.
    2. Simulate the environment as in Game 5.
    3. If $\mathcal{A}$ outputs $(y^*, \sigma^*_s)$ with $y^*$ not joined, output this as a forgery.
```

**Game 7: Extraction from $O^* = \pi^*_o$ (L2).** Form $x^*_o = (\mathsf{pk}_\Sigma, \mathsf{pk}^*_\Theta, c^*, m^*, y^*)$ and extract $\mathsf{sk}^*_\Theta$ from $\pi^*_o$. Check that $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1$. Abort on failure.

*Transition.* If $\pi^*_o$ verifies but extraction fails, SSE is broken. There exists $\mathcal{B}^{(4)}_{\mathsf{sse}}$ such that

$$|\Pr[S_7] - \Pr[S_6]| \leq \varepsilon^{(4)}_{\mathsf{sse}}(\lambda).$$

```
$\mathcal{B}^{(4)}_{\mathsf{sse}}(1^\lambda)$
    1. Receive $\mathsf{crs}^2_\Pi$ with extractor from the SSE challenger.
    2. Run $\mathcal{A}$ in the Game 6 environment.
    3. On an accepting $(x^*_o, \pi^*_o)$, forward it to the SSE challenger.
```

**Game 8: Decryption-Soundness check.** With $c^* = \mathsf{Enc}_\Theta(\mathsf{pk}^*_\Theta, y^*; r^*)$ and $\mathsf{KVfy}_\Theta(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta) = 1$, abort if $\mathsf{Dec}_\Theta(\mathsf{sk}^*_\Theta, c^*) \neq y^*$.

*Transition.* Otherwise Decryption-Soundness is violated. There exists $\mathcal{B}_{\mathsf{ds}}$ such that

$$|\Pr[S_8] - \Pr[S_7]| \leq \varepsilon_{\mathsf{ds}}(\lambda).$$

```
$\underline{\mathcal{B}_{\mathsf{ds}}(1^\lambda)}$   1. Receive $\mathsf{pp}_\Theta$ from the DS challenger.
                 2. Run $\mathcal{A}$ to obtain $(\mathsf{sk}^*_\Theta, \mathsf{pk}^*_\Theta, y^*, r^*)$.
                 3. Output this tuple to the DS challenger.
```

**Conclusion and Advantage Analysis.** We have shown that any successful adversary $\mathcal{A}$ against opening unforgeability can be turned into an adversary against one of the underlying assumptions. Concretely,

$$\Pr[S_0] \leq \varepsilon^{(1)}_{\mathsf{zk}} + \varepsilon^{(2)}_{\mathsf{zk}} + \varepsilon_{\mathsf{dsigunf}} + \varepsilon^{(3)}_{\mathsf{sse}} + \varepsilon^{(4)}_{\mathsf{sse}} + \varepsilon_{\mathsf{ds}}.$$

Since all terms are negligible, the scheme satisfies opening unforgeability. □

**Conclusion.** Any adversary that forges a valid opening for a non-joined pair $(\mathsf{opk}^*, \mathsf{upk}^*)$ must necessarily trigger one of the reductions above: either distinguish real vs. simulated $L_1$ or $L_2$ proofs (breaking ZK; Games 0→1 and 1→2), produce a verifying but non-extractable/inconsistent $L_1$ proof (breaking SSE; Game 4→5), yield a DSIG forgery when the claimed member was never joined (Game 5→6), produce a verifying but non-extractable/inconsistent $L_2$ proof (breaking SSE; Game 6→7), or violate Decryption Soundness (Game 7→8). Since each underlying advantage is negligible, $\Delta'$ satisfies unforgeability.

# E Proofs of Seperation

**Conventions.** All keys are consistently extended with *anchors* of fixed structure. In the honest case, these anchors are set to $\perp$. This ensures uniform tuple length and simplifies reductions.

Throughout, let $\Delta$ be a secure baseline group signature scheme that satisfies all eight notions. All constructions are black-box wrappers of $\Delta$.

Likewise, by abuse of notation, we always sign $m \overset{\text{def}}{=} (m, \mathsf{ipk}, \mathsf{opk})$ in the underlying $\Delta$ to bind the group public keys to the message. We do not make this explicit in the algorithms to have a crisp write-up. Wrapping and unwrapping for the underlying $\Delta$ is therefore not made explicit.

## E.1 Proof of Theorem 23 ($\boldsymbol{\Delta}^{\neg\mathsf{ANON}}$)

*Proof.* We provide a counterexample via the black-box wrapper of $\Delta$ given in Construction 5.

**Why anonymity fails.** In the left-or-right game, the challenge is $\sigma'_m = (\sigma^\star_m, \mathsf{upk}_b)$ with hidden bit $b \in \{0, 1\}$. The adversary reads $\mathsf{upk}_b$ from the cleartext trailer and outputs $b$ with advantage $\approx 1$. Thus $\mathsf{Anon}$ is violated.

**Why the other seven properties hold.**

- **Traceability (TRC)**: an unopenable valid $\sigma'_m$ maps to an unopenable valid $\sigma^\star_m$.
- **Trace-Soundness (TS)**: two distinct accepting openings for $(m, \sigma'_m)$ induce two for $(m', \sigma^\star_m)$.

$\underline{\mathsf{Sgn}'_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)}$**:** Let $m' \leftarrow (m, \mathsf{upk})$. Compute $\sigma^\star_m \leftarrow \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m')$. Return $\sigma'_m \leftarrow (\sigma^\star_m, \mathsf{upk})$.
$\underline{\mathsf{Vfy}'_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma'_m)}$**:** Parse $\sigma'_m = (\sigma^\star_m, \mathsf{upk})$ and set $m' \leftarrow (m, \mathsf{upk})$. Return $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m', \sigma^\star_m)$.
$\underline{\mathsf{Opn}'_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma'_m)}$**:** Parse $\sigma'_m = (\sigma^\star_m, \mathsf{upk})$ and set $m' \leftarrow (m, \mathsf{upk})$. Return $\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m', \sigma^\star_m)$.
$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma'_m, \mathsf{upk}^*, O)}$**:** Parse $\sigma'_m = (\sigma^\star_m, \mathsf{upk})$ and set $m' \leftarrow (m, \mathsf{upk})$. Return $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m', \sigma^\star_m, \mathsf{upk}^*, O)$.

Construction 5: Wrapper $\Delta^{\neg\mathsf{ANON}}$

- **Non-Frameability (NF)**): any frame on $\mathsf{upk}$ lifts to a frame in $\Delta$.
- **Non-Claimability (NC)**): accepted re-assignment on a logged transcript in the wrapper contradicts that in $\Delta$.
- **Certificate-Unforgeability (CUnf)**): forging a certificate/signature in the wrapper for $(m, \sigma'_m)$ forges one in $\Delta$ for $(m', \sigma^\star_m)$.
- **Key-Unforgeability (KUnf)**): a fresh valid $\sigma'_m$ yields a fresh valid $\sigma^\star_m$.
- **Opening-Unforgeability (OpU)**: an accepted opening to a never-joined identity in the wrapper lifts to $\Delta$.

□

**Conclusion.** $\Delta^{\neg\mathsf{ANON}}$ breaks anonymity with advantage $\approx 1$ while preserving Traceability, Trace-Soundness, Non-Frameability, Non-Claimability, Certificate-Unforgeability, Key-Unforgeability, and Opening-Unforgeability by black-box reduction to $\Delta$.

### E.2 Proof of Theorem 24 ($\boldsymbol{\Delta}^{\neg\mathsf{TS}}$)

*Proof.* We provide a counterexample via the black-box wrapper of $\Delta$ in Construction 6.

**Why trace-soundness fails.** In the $\mathsf{TS}$ game (Fig. 7) the adversary controls issuer and opener for both pairs. It sets for each chosen pair $(\mathsf{ipk}^*_i, \mathsf{opk}^*_i)$ the markers equal and non-$\perp$: $\sigma^\star_{I,i} = \sigma^\star_{O,i} = \sigma^\star$. Pick any $m$ and use the reserved signature $\sigma^\star$. Then $\mathsf{Vfy}'_\Delta = 1$ by the gadget with probability 1, since by construction no honest $\mathsf{Sgn}_\Delta$ invocation ever emits $\sigma^\star$. For the *same* $(m, \sigma^\star)$, output two distinct openings $(\mathsf{upk}_0, O_0) = (\mathtt{tag}_0, \perp)$ and $(\mathsf{upk}_1, O_1) = (\mathtt{tag}_1, \perp)$. Both are accepted by $\mathsf{Jdg}'_\Delta$ and $\mathsf{upk}_0 \neq \mathsf{upk}_1$, hence one signature admits two valid openings, violating trace-soundness.

**Why the other seven properties hold.**

- **Anonymity (ANON).** The opener is honest. Hence, distributions match $\Delta$; any distinguisher transfers.
- **Non-Frameability (NF).** The gadget accepts only tags in $\mathcal{T}$, disjoint from the member-ID domain; any successful frame must occur on the baseline branch, contradicting NF of $\Delta$.
- **Traceability (TRC).** The opener is honest, hence $\sigma^\star_O = \perp$; gadget off. Every verifying signature opens as in $\Delta$.

$\underline{\mathsf{KgI}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$. Fix a distinguished bitstring $\sigma^\star$ that is guaranteed not to occur as output of $\mathsf{Sgn}_\Delta$ (e.g., a reserved out-of-band string). Assume tags $\mathcal{T} := \{\mathtt{tag}_0, \mathtt{tag}_1\}$ disjoint from the member-ID domain. An honest issuer sets $\sigma^\star_I \leftarrow \perp$ and $\mathcal{T} \leftarrow \perp$; a corrupt issuer may choose any $\sigma^\star_I \neq \perp$ and set $\mathcal{T}$ arbitrarily. Publish $\mathsf{ipk}' \leftarrow (\mathsf{ipk}, \sigma^\star_I, \mathcal{T})$ and keep $\mathsf{isk}' \leftarrow \mathsf{isk}$.

$\underline{\mathsf{KgO}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$. An honest opener sets $\sigma^\star_O \leftarrow \perp$ and $\mathcal{T} \leftarrow \perp$; a corrupt opener may choose any $\sigma^\star_O \neq \perp$ and set $\mathcal{T}$ arbitrarily. Publish $\mathsf{opk}' \leftarrow (\mathsf{opk}, \sigma^\star_O, \mathcal{T})$ and keep $\mathsf{osk}' \leftarrow \mathsf{osk}$.

$\underline{\mathsf{Vfy}'_\Delta(\mathsf{ipk}', \mathsf{opk}', m, \sigma'_m)}$**:** Parse $\mathsf{ipk}' = (\mathsf{ipk}, \sigma^\star_I, \mathcal{T})$ and $\mathsf{opk}' = (\mathsf{opk}, \sigma^\star_O, \mathcal{T})$. If $\exists \mathtt{tag}_i \in \mathcal{T}$ which is a valid public key, return 0. If $(\sigma^\star_I = \sigma^\star_O \neq \perp) \wedge (\sigma'_m = \sigma^\star_I)$, return 1. Otherwise return $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma'_m)$.

$\underline{\mathsf{Opn}'_\Delta(\mathsf{osk}', \mathsf{ipk}', m, \sigma'_m)}$**:** Parse $(\sigma^\star_I, \sigma^\star_O)$. If $(\sigma^\star_I = \sigma^\star_O \neq \perp) \wedge (\sigma'_m = \sigma^\star)$, return $\perp$. Otherwise return $\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma'_m)$.

$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk}', \mathsf{opk}', m, \sigma'_m, \mathsf{upk}^*, O)}$**:** Parse $(\sigma^\star_I, \sigma^\star_O, \mathcal{T})$. If $(\sigma^\star_I = \sigma^\star_O \neq \perp) \wedge (\sigma'_m = \sigma^\star_I)$, then accept iff $\mathsf{upk}^* \in \mathcal{T}$. Otherwise return $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma'_m, \mathsf{upk}^*, O)$.

Construction 6: Wrapper $\Delta^{\neg\mathsf{TS}}$

- **Non-Claimability (NC).** Logged honest transcripts coincide with those of $\Delta$; any accepted re-assignment contradicts NC of $\Delta$.
- **Certificate-Unforgeability (CU).** The issuer is honest; gadget off on oracle outputs. Any minted valid transcript reduces to a CU break of $\Delta$.
- **Key-Unforgeability (KU).** With honest issuer and replay filter, any fresh valid $\sigma'_m$ maps to a fresh valid $\sigma^\star_m$ in $\Delta$.
- **Opening-Unforgeability (OpU).** With an honest issuer, the gadget is off on logged transcripts; any accepted opening for a never-joined identity reduces to an OpU break of $\Delta$.

□

**Conclusion.** $\Delta^{\neg\mathsf{TS}}$ preserves ANON, NF, TRC, NC, CU, KU, and OpU by black-box reduction to $\Delta$, yet admits two distinct accepting openings for the same signature under issuer–opener collusion, thus breaking trace-soundness.

### E.3 Proof of Theorem 25 ($\Delta^{\neg\mathsf{TRC}}$)

*Proof.* We provide a counterexample via the following black-box wrapper of $\Delta$.

$\underline{\mathsf{KgI}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$. An honest issuer sets a traceability flag $b_{\mathsf{trc}} \leftarrow 0$; a corrupt issuer may set $b_{\mathsf{trc}} \leftarrow 1$. Publish $\mathsf{ipk}' \leftarrow (\mathsf{ipk}, b_{\mathsf{trc}})$ and keep $\mathsf{isk}' \leftarrow \mathsf{isk}$.

$\underline{\mathsf{Opn}'_\Delta(\mathsf{osk}', \mathsf{ipk}', m, \sigma_m)}$**:** Parse $\mathsf{ipk}' = (\mathsf{ipk}, b_{\mathsf{trc}})$. If $b_{\mathsf{trc}} = 1$ return $\perp$. Otherwise return $\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m)$.

$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk}', \mathsf{opk}, m, \sigma_m, \mathsf{upk}^*, O)}$**:** Parse $\mathsf{ipk}' = (\mathsf{ipk}, b_{\mathsf{trc}})$. If $b_{\mathsf{trc}} = 1$ return 0. Otherwise return $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}^*, O)$.

Construction 7: Wrapper $\Delta^{\neg\mathsf{TRC}}$ disabling opening under a corrupt issuer

**Why traceability fails.** In the $\mathsf{TRC}$ game (Fig. 5) the adversary controls the issuer and publishes $b_{\mathsf{trc}} = 1$. It uses any member key to produce a valid baseline signature $\sigma_m$ on $m$, so $\mathsf{Vfy}'_{\Delta} = 1$. By construction, $\mathsf{Opn}'_{\Delta}$ returns $\perp$ and $\mathsf{Jdg}'_{\Delta}$ returns $0$ on that transcript. Thus there exists a valid but unopenable signature, violating traceability.

**Why the other seven properties hold.**

- **Anonymity (ANON), Fig. 4.** Challenge signatures come from the signing oracle; the adversary never obtains openings on them anyway. Suppressing extra openings leaks no additional information, so distributions match $\Delta$.
- **Non-Frameability (NF).** Framing requires passing $\mathsf{Jdg}'_{\Delta}$. In the gadget branch ($b_{\mathsf{trc}} = 1$) it always returns $0$, so any successful frame must occur in the baseline branch ($b_{\mathsf{trc}} = 0$) and contradict NF of $\Delta$.
- **Trace-Soundness (TS).** If $b_{\mathsf{trc}} = 1$, no opening is accepted, hence two different accepted openings cannot exist. If $b_{\mathsf{trc}} = 0$, we inherit TS from $\Delta$.
- **Non-Claimability (NC).** NC fixes an honest transcript, so $b_{\mathsf{trc}} = 0$ there; any reassignment would already break NC of $\Delta$.
- **Certificate-Unforgeability (CUnf).** In CU the issuer is honest, i.e. $b_{\mathsf{trc}} = 0$; gadget off. Any forged certificate or signature reduces to CU of $\Delta$.
- **Key-Unforgeability (KUnf).** Same: the issuer is honest, so gadget off. Any fresh valid signature reduces to KU of $\Delta$.
- **Opening-Unforgeability (OpU).** With an honest issuer ($b_{\mathsf{trc}} = 0$), gadget off. Any accepted opening to a never-joined identity contradicts OpU of $\Delta$.

□

**Conclusion.** $\Delta^{\neg\mathsf{TRC}}$ preserves ANON, NF, TS, NC, CUnf, KUnf, and OpU by straight-line black-box reductions to $\Delta$, yet produces valid signatures that are unopenable whenever the issuer's flag is set to $1$, thereby breaking traceability.

### E.4 Proof of Theorem 26 ($\boldsymbol{\Delta}^{\neg\mathsf{NF}}$)

*Proof.* We provide a counterexample in Construction 8.

**Why non-frameability fails.** In the NF game the adversary controls issuer and opener. It sets $y_I^\star = y_O^\star = y^{\text{hon}}$, the identity of the single honest member. It then forges any valid baseline signature $\sigma_m$ for $(m, 1)$. Verification accepts it by design, although no honest transcript for $(m, 1)$ exists. In $\mathsf{Jdg}'_{\Delta}$, the anchors match and force acceptance of the claim $\mathsf{upk}^* = y^{\text{hon}}$, even though this honest user never signed. Thus, a frame succeeds.

**Why the other seven properties hold.**

- **Anonymity (ANON).** Challenge signatures are always generated on $(m, 0)$ with honest authorities, anchors $\perp$; distribution matches $\Delta$.
- **Traceability (TRC).** In the TRC game the opener is honest, so $y_O^\star = \perp$; gadget off, reducing to the baseline.
- **Trace-Soundness (TS).** For any transcript in the gadget branch, the only admissible opening is $y^\star$; otherwise TS reduces to the baseline. Hence no two distinct valid openings exist.

$\underline{\mathsf{KgI}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$. An honest issuer sets $y_I^\star \leftarrow \bot$; a corrupt issuer may choose any $y^\star$. Publish $\mathsf{ipk}' = (\mathsf{ipk}, y_I^\star)$ and keep $\mathsf{isk}' = \mathsf{isk}$.

$\underline{\mathsf{KgO}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$. An honest opener sets $y_O^\star \leftarrow \bot$; a corrupt opener may choose any $y^\star$. Publish $\mathsf{opk}' = (\mathsf{opk}, y_O^\star)$ and keep $\mathsf{osk}' = \mathsf{osk}$.

$\underline{\mathsf{Sgn}'_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}', \mathsf{opk}', m)}$**:** Form $m^\circ \leftarrow (m, 0)$ and compute

$$\sigma_m \leftarrow \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m^\circ).$$

Return $\sigma_m$.

$\underline{\mathsf{Vfy}'_\Delta(\mathsf{ipk}', \mathsf{opk}', m, \sigma_m)}$**:** Accept if

$$\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, (m, 0), \sigma_m) = 1 \quad \vee \quad \mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, (m, 1), \sigma_m) = 1.$$

$\underline{\mathsf{Opn}'_\Delta(\mathsf{osk}', \mathsf{ipk}', m, \sigma_m)}$**:** If verification succeeded on $(m, 0)$, call the baseline opener. If verification succeeded only on $(m, 1)$, return $\bot$.

$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk}', \mathsf{opk}', m, \sigma_m, \mathsf{upk}^*, O)}$**:** Parse anchors $(y_I^\star, y_O^\star)$. If $y_I^\star = y_O^\star \neq \bot$ and $\mathsf{Vfy}'_\Delta(\mathsf{ipk}', \mathsf{opk}', (m, 1), \sigma_m) = 1$, then accept iff $\mathsf{upk}^* = y_I^\star$ (ignoring $O$). Otherwise, return $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m^\circ, \sigma_m, \mathsf{upk}^*, O)$ for $m^\circ = (m, 0)$.

Construction 8: Wrapper $\Delta^{\neg\mathsf{NF}}$

- **Non-Claimability (NC).** Honest transcripts use $(m, 0)$, where the bypass is inert. Any accepted reassignment reduces to NC of $\Delta$.
- **Certificate-Unforgeability (CUnf).** In the CUnf game the issuer is honest, so $y_I^\star = \bot$; the gadget is off. Any forged certificate already contradicts baseline CUnf.
- **Key-Unforgeability (KUnf).** In the KUnf game the issuer is honest, so $y_I^\star = \bot$; the gadget is off. Any valid fresh signature reduces to a KUnf break of $\Delta$.
- **Opening-Unforgeability (OpU).** With an honest issuer in the OpU game, the gadget is off on logged transcripts; any accepted opening for a never-joined identity contradicts baseline OpU.

□

**Conclusion.** $\Delta^{\neg\mathsf{NF}}$ breaks non-frameability by forcing an opening to an anchored honest identity when issuer and opener collude, while preserving ANON, TRC, TS, NC, CUnf, KUnf, and OpU by straight-line black-box reduction to $\Delta$.

### E.5 Proof of Theorem 27 ($\boldsymbol{\Delta^{\neg\mathsf{NC}}}$)

*Proof.* We provide a counter-example in Construction 9.

**Why non-claimability fails (issuer and opener corrupt).** In the NC game the adversary corrupts *both* issuer and opener and sets a common anchor $\mathsf{app}_G = \mathsf{app}_O = r \neq \bot$, which fixes $(\mathsf{usk}^\dagger, \mathsf{upk}^\dagger) \leftarrow \mathsf{KgM}_\Delta(r)$. Let $(m, \sigma')$ be any honestly logged transcript output by the signing oracle of the single honest member $\mathsf{upk} \in \mathcal{Y}$. Since the opener is corrupt, it computes a baseline witness $\widetilde{O}$ with $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma', \mathsf{upk}, \widetilde{O}) = 1$ and outputs

$$(\mathsf{upk}^*, O) \;\leftarrow\; \big(\mathsf{upk}^\dagger,\; (\widehat{\mathsf{upk}}, \widetilde{O})\big) \quad \text{with} \quad \widehat{\mathsf{upk}} = \mathsf{upk}.$$

$\underline{\mathsf{KgI}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{KgI}_\Delta(\mathsf{pp}_\Delta)$. An honest issuer sets $\mathsf{app}_G \leftarrow \bot$; a corrupt issuer may choose any randomness $r_G$ and set $\mathsf{app}_G \leftarrow r_G$. Publish $\mathsf{ipk}' \leftarrow (\mathsf{ipk}, \mathsf{app}_G)$ and keep $\mathsf{isk}' \leftarrow \mathsf{isk}$.

$\underline{\mathsf{KgO}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$. An honest opener sets $\mathsf{app}_O \leftarrow \bot$; a corrupt opener may choose any randomness $r_O$ and set $\mathsf{app}_O \leftarrow r_O$. Publish $\mathsf{opk}' \leftarrow (\mathsf{opk}, \mathsf{app}_O)$ and keep $\mathsf{osk}' \leftarrow \mathsf{osk}$.

$\underline{\mathsf{Vfy}'_\Delta(\mathsf{ipk}', \mathsf{opk}', m, \sigma')}$**:** Parse $(\mathsf{ipk}', \mathsf{opk}') = ((\mathsf{ipk}, \mathsf{app}_G), (\mathsf{opk}, \mathsf{app}_O))$ and return $\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma')$.

$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk}', \mathsf{opk}', m, \sigma', \mathsf{upk}^*, O)}$**:** Parse $(\mathsf{ipk}', \mathsf{opk}') = ((\mathsf{ipk}, \mathsf{app}_G), (\mathsf{opk}, \mathsf{app}_O))$. If $\mathsf{app}_G = \mathsf{app}_O = r \neq \bot$, then compute $(\mathsf{usk}^\dagger, \mathsf{upk}^\dagger) \leftarrow \mathsf{KgM}_\Delta(r)$ and parse $O = (\widehat{\mathsf{upk}}, \widetilde{O})$. Accept iff

$$\mathsf{Vfy}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma') = 1 \ \wedge \ \mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma', \widehat{\mathsf{upk}}, \widetilde{O}) = 1 \ \wedge \ \mathsf{upk}^* = \mathsf{upk}^\dagger.$$

Otherwise, return $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma', \mathsf{upk}^*, O)$.

Construction 9: Wrapper $\Delta^{\neg\mathsf{NC}}$ (NC separation via randomness anchor)

By construction $\mathsf{Jdg}'_\Delta$ ignores claims to any identity except $\mathsf{upk}^\dagger$, checks the baseline witness against the embedded true identity $\widehat{\mathsf{upk}}$, and accepts exactly when $\mathsf{upk}^* = \mathsf{upk}^\dagger$. The key $\mathsf{upk}^\dagger$ never joined, yet the claim is accepted. Hence NC is violated.

**Why the other seven properties hold.**

- **Anonymity (ANON).** In the ANON experiment the opener is honest, hence $\mathsf{app}_O = \bot$ and the wrapper is inactive. Challenge signatures are generated and verified exactly as in $\Delta$; distributions match those of the baseline scheme.
- **Non-Frameability (NF).** In NF only the user is honest; issuer and opener are corrupt. Frames target an honest joined identity in $\mathcal{Y}$. The gate accepts *only* claims to the adversary-fixed $\mathsf{upk}^\dagger = \mathsf{KgM}_\Delta(r)$; it never accepts a claim to the honest user. Thus any frame would already break NF of $\Delta$.
- **Traceability (TRC).** In TRC the opener is honest, hence $\mathsf{app}_O = \bot$ and the gate is off. Every verifying signature opens as in $\Delta$.
- **Trace-Soundness (TS).** For fixed $(\mathsf{ipk}', \mathsf{opk}', m, \sigma')$ the gate, when active, accepts at most one identity ($\mathsf{upk}^\dagger$) and rejects all others; otherwise TS reduces to $\Delta$. Thus no two distinct accepted openings exist.
- **Certificate-Unforgeability (CUnf).** In CUnf the issuer is honest, so $\mathsf{app}_G = \bot$; gate off. Any forged certificate contradicts CUnf of $\Delta$.
- **Key-Unforgeability (KUnf).** In KUnf the issuer is honest; gate off. Any fresh valid signature reduces to a KUnf break of $\Delta$.
- **Opening-Unforgeability (OpU).** With an honest issuer the gate is off on logged transcripts. Any accepted opening to a never-joined valid identity would already violate OpU of $\Delta$.

□

**Conclusion.** $\Delta^{\neg\mathsf{NC}}$ preserves ANON, NF, TRC, TS, CUnf, KUnf, and OpU by straight-line reduction to $\Delta$, yet lets a corrupt issuer–opener pair *relabel* any honest transcript to the anchor-derived key $\mathsf{upk}^\dagger$ from $\mathsf{KgM}_\Delta(r)$, thereby breaking NC.

### E.6 Proof of Theorem 28 ($\boldsymbol{\Delta}^{\neg\mathsf{CUnf}}$)

*Proof.* We provide a counter-example in Construction 10 that mirrors the KUnf separation, replacing $h = f(\sigma_s)$ by $h = f(\mathsf{usk})$.

$\underline{\mathsf{KgO}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$. An honest opener sets $b_{\mathsf{cunf}} \leftarrow 0$; a corrupt opener may set $b_{\mathsf{cunf}} \leftarrow 1$. Publish $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{cunf}})$ and keep $\mathsf{osk}' = \mathsf{osk}$.

$\underline{\mathsf{Sgn}'_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}', m)}$**:** Parse $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{cunf}})$. If $b_{\mathsf{cunf}} = 0$, compute $\sigma_m \leftarrow \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)$ and return $\sigma_m$. If $b_{\mathsf{cunf}} = 1$, compute $h \leftarrow f(\mathsf{usk})$ for a one-way function $f$ and return $\sigma_m \leftarrow \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, h)$.

$\underline{\mathsf{Opn}'_\Delta(\mathsf{osk}', \mathsf{ipk}, m, \sigma_m)}$**:** Parse $\mathsf{osk}' = \mathsf{osk}$ and $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{cunf}})$. If $b_{\mathsf{cunf}} = 0$, return $\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m)$. Else, return $\perp$.

$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk}, \mathsf{opk}', m, \sigma_m, \mathsf{upk}^*, O)}$**:** Parse $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{cunf}})$. If $b_{\mathsf{cunf}} = 0$, return $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}^*, O)$. If $b_{\mathsf{cunf}} = 1$, return 0.

Construction 10: Wrapper $\Delta^{\neg\mathsf{CUnf}}$

**Why certificate-unforgeability fails.** In the CUnf game the issuer is honest and the adversary chooses the opener key. Set $b_{\mathsf{cunf}} = 1$ in $\mathsf{opk}'$. Query $\mathcal{O}_{\mathsf{KgM}_\Delta}$ to obtain $(\mathsf{usk}, \mathsf{upk})$. Compute $h \leftarrow f(\mathsf{usk})$ and query the signing oracle once on some $m \neq h$ to receive $\sigma_m$ (which is in fact a signature on $h$ under the gadget). Output the forgery $(\sigma^*_m, m^*, \mathsf{opk}^*)$ with $m^* = h$ and the same $\sigma^*_m$. Verification under the baseline check succeeds (the signature is on $h$), while the replay filter does not trigger since the logged oracle tuple used $m \neq h$. Thus a fresh valid signature is minted without any join-certificate issued by $\mathcal{O}_{\mathsf{Join}_\Delta}$, violating certificate-unforgeability.

**Why the other seven properties hold.**

- **Anonymity (ANON).** Challenge signatures use an honest opener ($b_{\mathsf{cunf}} = 0$); distributions match $\Delta$.
- **Non-Frameability (NF).** A frame requires an accepted opening. For $b_{\mathsf{cunf}} = 1$, $\mathsf{Jdg}'_\Delta$ always rejects; any successful frame would contradict NF of $\Delta$ on the baseline branch.
- **Traceability (TRC).** In TRC the opener is honest ($b_{\mathsf{cunf}} = 0$); every verifying signature is openable as in $\Delta$.
- **Trace-Soundness (TS).** In the gadget branch $\mathsf{Jdg}'_\Delta$ always rejects; in the baseline branch TS follows from $\Delta$.
- **Non-Claimability (NC).** The NC experiment fixes an honestly logged transcript with an honest opener ($b_{\mathsf{cunf}} = 0$); any accepted reassignment would already break NC of $\Delta$.
- **Key-Unforgeability (KUnf).** The gadget only changes what message honest signers use when $b_{\mathsf{cunf}} = 1$; it grants no extra power to produce valid group signatures without member secrets. A KUnf break would already break $\Delta$.
- **Opening-Unforgeability (OpU).** With an honest issuer and $b_{\mathsf{cunf}} = 0$, transcripts reduce to the baseline; any accepted opening to a never-joined identity would contradict OpU of $\Delta$.

□

### E.7 Proof of Theorem 29 ($\boldsymbol{\Delta}^{\neg\mathsf{KUnf}}$)

*Proof.* We provide a counter-example in Construction 11.

$\underline{\mathsf{KgO}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$. An honest opener sets $b_{\mathsf{kunf}} \leftarrow 0$; a corrupt opener may set $b_{\mathsf{kunf}} \leftarrow 1$. Publish $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{kunf}})$ and keep $\mathsf{osk}' = \mathsf{osk}$.

$\underline{\mathsf{Sgn}'_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}', m)}$**:** Parse $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{kunf}})$. If $b_{\mathsf{kunf}} = 0$, compute $\sigma_m \leftarrow \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, m)$ and return $\sigma_m$. If $b_{\mathsf{kunf}} = 1$, compute $h \leftarrow f(\sigma_s)$ for a one-way function $f$ and return $\sigma_m \leftarrow \mathsf{Sgn}_\Delta(\mathsf{usk}, \sigma_s, \mathsf{ipk}, \mathsf{opk}, h)$.

$\underline{\mathsf{Opn}'_\Delta(\mathsf{osk}', \mathsf{ipk}, m, \sigma_m)}$**:** Parse $\mathsf{osk}' = \mathsf{osk}$ and $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{kunf}})$. If $b_{\mathsf{kunf}} = 0$, return $\mathsf{Opn}_\Delta(\mathsf{osk}, \mathsf{ipk}, m, \sigma_m)$. Else, return $\perp$.

$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk}, \mathsf{opk}', m, \sigma_m, \mathsf{upk}^*, O)}$**:** Parse $\mathsf{opk}' = (\mathsf{opk}, b_{\mathsf{kunf}})$. If $b_{\mathsf{kunf}} = 0$, return $\mathsf{Jdg}_\Delta(\mathsf{ipk}, \mathsf{opk}, m, \sigma_m, \mathsf{upk}^*, O)$. If $b_{\mathsf{kunf}} = 1$, return $0$.

Construction 11: Wrapper $\Delta^{\neg\mathsf{KUnf}}$

**Why key-unforgeability fails.** In the KUnf game the adversary corrupts the opener and sets $b_{\mathsf{kunf}} = 1$. It obtains join-certificates $\sigma_s$ via $\mathcal{O}_{\mathsf{Join}_\Delta}$, computes $h \leftarrow f(\sigma_s)$, and queries the signing oracle once on some $m \neq h$ to receive $\sigma_m$ (which is in fact a signature on $h$). It then outputs the forgery $(\sigma_m^*, m^*, \mathsf{opk}^*)$ with $m^* = h$ and the same $\sigma_m^*$. Verification under the baseline check succeeds (the signature is on $h$), while the replay filter does not trigger since the logged oracle tuple used $m \neq h$. Thus the adversary mints a valid signature without any member secret, violating KUnf.

**Why the other seven properties hold.**

- **Anonymity (ANON).** Challenge signatures use an honest opener ($b_{\mathsf{kunf}} = 0$); distributions match $\Delta$.
- **Non-Frameability (NF).** A frame requires an accepted opening. For $b_{\mathsf{kunf}} = 1$, $\mathsf{Jdg}'_\Delta$ always rejects; any successful frame would contradict NF of $\Delta$ on the baseline branch.
- **Traceability (TRC).** In TRC the opener is honest ($b_{\mathsf{kunf}} = 0$); every verifying signature is openable as in $\Delta$.
- **Trace-Soundness (TS).** In the gadget branch $\mathsf{Jdg}'_\Delta$ always rejects; in the baseline branch we inherit TS from $\Delta$.
- **Non-Claimability (NC).** The NC experiment fixes an honestly logged transcript with an honest opener ($b_{\mathsf{kunf}} = 0$); any accepted re-assignment would already break NC of $\Delta$.
- **Certificiate Unforgeability (CUnf).** Certificates are issued by the honest issuer as in $\Delta$; the wrapper only causes extra signatures on $f(\sigma_s)$ of already-issued certificates. Forging a certificate still contradicts CUnf of $\Delta$.
- **Opening-Unforgeability (OpU).** With an honest issuer and $b_{\mathsf{kunf}} = 0$, transcripts reduce to the baseline; any accepted opening to a never-joined identity would contradict OpU of $\Delta$.

□

**Conclusion.** The wrapper $\Delta^{\neg\mathsf{KUnf}}$ preserves ANON, NF, TRC, TS, NC, CUnf, and OpU by straight-line reduction to $\Delta$, yet it breaks Key-Unforgeability: a corrupt opener can leverage the hash of honestly issued join-certificates to mint valid signatures without any member secret key.

### E.8 Proof of Theorem 30 ($\boldsymbol{\Delta}^{\neg\mathsf{OpU}}$)

*Proof.* We provide a counter-example in Construction 12.

$\underline{\mathsf{KgO}'_\Delta(\mathsf{pp}_\Delta)}$**:** Run $(\mathsf{osk},\mathsf{opk}) \leftarrow \mathsf{KgO}_\Delta(\mathsf{pp}_\Delta)$. An honest opener sets $\mathsf{app} \leftarrow \bot$. A corrupt opener may choose any randomness $r$ and set $\mathsf{app} \leftarrow r$. Publish $\mathsf{opk}' \leftarrow (\mathsf{opk},\mathsf{app})$ and keep $\mathsf{osk}' \leftarrow \mathsf{osk}$.

$\underline{\mathsf{Vfy}'_\Delta(\mathsf{ipk},\mathsf{opk}',m,\sigma')}$**:** Parse $\mathsf{opk}' = (\mathsf{opk},\mathsf{app})$. Return $\mathsf{Vfy}_\Delta(\mathsf{ipk},\mathsf{opk},(m,0),\sigma') \vee \mathsf{Vfy}_\Delta(\mathsf{ipk},\mathsf{opk},(m,1),\sigma')$.

$\underline{\mathsf{Jdg}'_\Delta(\mathsf{ipk},\mathsf{opk}',m,\sigma',\mathsf{upk}^*,O)}$**:** Parse $\mathsf{opk}' = (\mathsf{opk},\mathsf{app})$. If $\mathsf{app} = r \neq \bot$ and $\mathsf{Vfy}_\Delta(\mathsf{ipk},\mathsf{opk},(m,1),\sigma') = 1$, then set $(\mathsf{usk}^\dagger,\mathsf{upk}^\dagger) \leftarrow \mathsf{KgM}_\Delta(r)$ and accept iff

$$\mathsf{upk}^* = \mathsf{upk}^\dagger,$$

ignoring $O$. Otherwise return $\mathsf{Jdg}_\Delta(\mathsf{ipk},\mathsf{opk},(m,0),\sigma',\mathsf{upk}^*,O)$.

Construction 12: Wrapper $\Delta^{\neg\mathsf{OpU}}$

**Why opening-unforgeability fails.** In the OpU game the issuer is honest, but the adversary controls the opener. It publishes some $r$, fixing $(\mathsf{usk}^\dagger,\mathsf{upk}^\dagger) \leftarrow \mathsf{KgM}_\Delta(r)$. Obtain any valid signature $\sigma'$ on $(m,1)$ from a (corrupted) joined member. Output $(\mathsf{upk}^*,O) = (\mathsf{upk}^\dagger,\bot)$. Then $\mathsf{Vfy}'_\Delta(\cdot) = 1$ and $\mathsf{Jdg}'_\Delta$ accepts an opening to $\mathsf{upk}^\dagger$, which never joined. Hence OpU is violated.

**Why the other seven properties hold.**

- **Anonymity (ANON).** Challenge signatures use an honest opener ($r = \bot$); distributions match those of $\Delta$.
- **Non-Frameability (NF).** A frame would require some $r$ with $\mathsf{KgM}_\Delta(r) = (\cdot,\mathsf{upk}_i)$ for an honest, and joined, $\mathsf{upk}_i$. Recovering such an $r$ from $\mathsf{upk}_i$ is infeasible; NF reduces to that of $\Delta$.
- **Traceability (TRC).** Opener honest ($r = \bot$); the wrapper is inactive; tracing as in $\Delta$.
- **Trace-Soundness (TS).** For a fixed $(m,\sigma')$ with flag 1 the wrapper names at most one identity ($\mathsf{upk}^\dagger$); otherwise TS reduces to $\Delta$.
- **Non-Claimability (NC).** In the NC experiment the opener may be corrupt, but the challenge transcript is always generated honestly and thus carries flag 0. Since the wrapper remaps only signatures with flag 1, it is inactive in the NC challenge. Any successful reassignment therefore reduces to NC of $\Delta$.
- **Certificate-Unforgeability (CUnf).** Signing remains unchanged; any certificate forgery already breaks CUnf of $\Delta$.

- **Key-Unforgeability (KUnf).** Sign/verify remain unchanged beyond openings; any fresh valid signature reduces to a KUnf break of $\Delta$.

$\square$

**Conclusion.** $\Delta^{\neg\mathsf{OpU}}$ preserves ANON, NF, TRC, TS, NC, CUnf, and KUnf by reduction to $\Delta$, yet for flag 1 accepts an opening to $\mathsf{upk}^{\dagger}$ from $\mathsf{KgM}_{\Delta}(r)$ exactly when the claimed key equals that value, with $r$ fixed by the corrupt opener and never joined. Hence OpU is violated.

## F On the Usage of AI

As required by recent AI Tool Policies (e.g., ACM's guidance adopted by several IACR venues), we disclose *what* AI tools were used in preparing this manuscript, *how* they were used, and *how we ensured* the integrity of the work. We (the authors) accept full responsibility for all content.

**AI-Tool Disclosure.** We used a generative AI assistant (OpenAI ChatGPT, model GPT-5 Thinking) for language polishing and LaTeX boilerplate. Scientific ideas, definitions, theorems, and proofs are by the authors. All AI-assisted output was reviewed and edited by the authors. No automated proof tools, code execution, or web search via the assistant were used.

**Summary.** We used the AI as a virtual reviewer and writing assistant. We used the AI only to help generate comments on the body of work (in particular for consistency and understandability), to review/polish language and boilerplate, and for LaTeX typesetting. All ideas are ours. The AI was not used to generate any ideas, proofs, or claims.

### F.1 Tools, Versions, and Access Modality

- **Tool:** OpenAI ChatGPT (model: GPT-5 Thinking)
- **Scope of use:** Interactive drafting and editing of LaTeX prose; help with wording, structure, and stylistic consistency; generation of LaTeX boilerplate for experiments, figures, and environments; red-teaming language for definitions and security games; and consistency checks for notation and quantifiers.
- **Non-use:** No code execution, no empirical evaluation, no automated proof assistants, and no external web browsing via the tool were used to generate any scientific claims in this paper. All citations and attributions were curated and verified by the authors.

### F.2 Summary of AI Involvement (High Level)

The AI assistant was used *as a writing and presentation aid* to accelerate drafting of sections whose scientific content we designed. Concretely, we supplied the mathematical intent, drafts, experiments, definitions, model choices, and security goals. The assistant proposed comments and LaTeX snippets. We iteratively revised, corrected, and, where necessary, rewrote the generated material before incorporating it into the body. In all cases, *the technical content, definitions, theorems, and proofs are our responsibility*.

### F.3 Granular Disclosure by Artifact

Below we describe the kinds of AI assistance per artifact type. In each item, “we” refers to the authors’ active role; “assistant” refers to the AI tool.

**Expository prose.** We used the assistant for first-pass phrasing and later polishing of expository paragraphs (motivation, overview, and “story-of-the-proof” descriptions), especially to remove ambiguity and reduce passive voice. We reviewed each sentence for correctness and alignment with our intended meaning.

**Formal definitions and experiments.** We provided the formal intent, interfaces, and structure; the assistant helped: (i) render experiments in a consistent tabbing/tabular LaTeX style; (ii) enforce consistent naming ($\mathcal{A}$, $\lambda$, $\mathcal{Q}$, oracle names); (iii) harmonize notation across preliminaries and the group-signature sections (e.g., instance vs. statement variables). All comments by the AI on the formal definitions and security experiments were verified line by line and, where appropriate, changed by the authors before being incorporated into the body.

**Contrived construction section.** We supplied the counterexample construction skeletons, threat model, and acceptance conditions. The assistant drafted an initial natural-language exposition and LaTeX structure, which we reworked; we validated every claim and supplied (or corrected) all proof obligations.

**Secure construction section.** We used the assistant to produce boilerplate for the language definitions $L_1, L_2$ and game-hopping skeletons. We authored the concrete statements, the precise game transitions, and all references to assumptions (ZK, SSE, statement binding, key verifiability, decryption soundness). The AI was used to help with the presentation and to increase consistency.

**Figures and pseudo-code environments.** The assistant produced LaTeX layout (e.g., `adjustbox`, `tabbing` blocks, captions). We ensured the semantics (inputs/outputs, abort conditions, and checks) are correct and consistent with the text.

**Consistency passes.** We used the assistant to flag common pitfalls, e.g., mismatched quantifiers, swapped roles (issuer vs. opener), inconsistent use of $\perp$, and silent dependence of statements on parameters. We decided on fixes and edited the text ourselves.

**Negative use-cases.** We did not ask the tool to produce its own proofs nor to generate unverifiable claims. No AI-generated proof was adopted; drafts were discarded. We did not use auto-citation or auto-bibliography features. All references were added and checked by the authors.

### F.4 What the AI Did *Not* Do

- It did not originate research questions, constructions, or security notions. Those are ours. The AI was only used to provide additional comments and to challenge our ideas and write-up.
- It did not perform any proofs independently. All proofs originate from the authors; assistant feedback was integrated only if correct.
- It did not search the web or compile related work. All literature claims and citations were decided by the authors.
- It did not run experiments, verify code, or produce empirical results.

### F.5 Quality Control and Risk Mitigation

Generative systems can introduce subtle errors, even if used only for checking (e.g., incorrect quantifiers, inconsistent experiments, or misplaced assumptions). We applied the following controls:

1. **Manual proof audit.** Every definition, experiment, and game-hop comment by the AI was checked by at least one author. We verified that each transition has a named assumption/reduction and appropriate success events.
2. **Assumption hygiene.** Each use of ZK, SSE, statement-binding, key-verifiability, and decryption-soundness is explicitly tied to a hop. We confirmed no assumption is used to prove itself (no circularity).

### F.6 Provenance, Reproducibility, and Transparency

To align with emerging policy expectations:

- **Granularity of disclosure.** Sections where the assistant's phrasing materially influenced the exposition (but not the underlying claims) are: parts of Figure 7 caption text, the contrived construction narrative and the "How to sign / open as a corrupt opener" exposition, as well as the security-game explanations. Final statements and proofs reflect the authors' edits and acceptance.
- **Attribution and authorship.** The assistant is *not* an author, provided no originality claims, and carries no responsibility for the work.

### F.7 Data, Ethics, and Privacy

- **No personal data.** We did not provide personal or sensitive data to the assistant.
- **No proprietary datasets.** The paper is theoretical. No datasets were uploaded or generated.
- **Plagiarism avoidance.** We treated AI output as comments on our work ("virtual reviewer") and as an interactive writing assistant. We used the AI output as needed and ensured originality of expression for our contributions. All citations are inserted by the authors.

### F.8 Limitations of AI Assistance

Even when used carefully, AI-generated drafts may (i) elide edge-case conditions, (ii) unintentionally change a quantifier or oracle domain, or (iii) suggest plausible but inapplicable references. Our mitigations (audit steps above) specifically targeted these failure modes. Any remaining errors are ours.

### F.9 Policy-Facing Statement

In line with current AI Tool Policies:

We used a generative AI assistant (OpenAI ChatGPT, model "GPT-5 Thinking") for language polishing, LaTeX boilerplate, and drafting of expository text and experiment layouts under author supervision. The scientific ideas, formal definitions, theorems, and proofs are the authors' work. We verified every AI-assisted passage for correctness and clarity, incorporated material where appropriate, and accept full responsibility for the content.

**Representative prompts (examples).**

1. **AI-usage disclosure paragraph (be honest).**
   Draft a 120–160 word "AI Tool Disclosure" paragraph for an LNCS paper. Be explicit that the assistant helped only with language polishing and LaTeX boilerplate, not with ideas, proofs, experiments, or citations. State that this paragraph itself was drafted by the assistant at the authors' request, and that the authors reviewed, edited, and accept full responsibility. Neutral academic tone; no hype; no policy advice.
2. **Language polish without content changes.**
   Polish the following paragraph for clarity and concision. Do not change any technical claims, symbols, or notation. Keep math and identifiers verbatim. Output only the revised paragraph.
   Text: <<<PARAGRAPH>>>
3. **LaTeX boilerplate for side-by-side experiments.**
   Produce LaTeX for two top-aligned boxes (minipage + adjustbox) placed side by side, compatible with `llncs` (no `caption`/`subcaption`). Left box title: <<<LEFT TITLE>>>; right box title: <<<RIGHT TITLE>>>. Each box contains a compact `tabbing` block. Thin frame, tight spacing, no vertical stretch.
4. **Consistency review (virtual reviewer).**
   Review the excerpt only for consistency: quantifier order, notation ($\mathcal{A}$, $\lambda$, $\mathcal{Q}$ sets), oracle interfaces, and replay-filter conditions. Report issues as a bullet list with one-line justifications. No rewrites, no new content.
   Excerpt: <<<TEXT>>>
5. **Red-team the security experiment.**
   Stress-test this experiment for likely gaps: missing abort conditions, insufficient replay filter (e.g., issuer key omission), unbound handles (certificate index, member key), and edge cases (unregistered keys, cross-context signing). Return a checklist (Yes/No + one sentence) and point to exact lines.
   Experiment: <<<CODE>>>

### F.10 Why We Included This Section

The point of this disclosure is twofold: (i) to give readers and reviewers a precise map of where AI assistance touches the text (presentation-level) versus where human authorship is essential (idea and proof-level), and (ii) to make explicit the checks we performed so that trust in the technical claims derives from transparent process and verifiable reasoning rather than from tooling.