Practical Adaptor Signatures for NP from Online/Offline NIZK*

Masayuki Abe[1], Dung Bui[2], Kelong Cong[3], Miyako Ohkubo[4], Zehua Shang[5], Akira Takahashi[6], and Mehdi Tibouchi[1]

[1] NTT Social Informatics Laboratories, Japan
[2] Northwestern University, USA
[3] Zama, France
[4] NICT, Japan
[5] Kyoto University & Yokohama National University, Japan
[6] J.P. Morgan, USA

Abstract. Adaptor signatures enable *conditional payments* on blockchain networks: a pre-signature is bound to a public instance Y and can be completed into a valid signature by any party knowing a witness y with $\mathcal{R}(Y, y) = 1$, while the pre-signature and full signature together allow the signer to extract y. Existing constructions only support specific algebraically-structured relations, lose witness privacy, or have significant overhead both in signature size and computation.

We present a practical adaptor signature framework for arbitrary NP relations, built from a new primitive we call *Online/Offline NIZK* whose proof is separated into online and offline parts. The key property is that the witness can be extracted from the online part knowing the randomness used in the offline part. We give two generic constructions: a *basic* scheme achieving witness hiding, and an *instance-hiding* scheme achieving a new, stronger forward-secure privacy notion we introduce.

We instantiate our framework for the AES-128 relation using the VOLE-in-the-Head paradigm and the FAEST post-quantum signature scheme. The resulting complete signature is 11.6 kB—roughly 60% smaller than the state-of-the-art adaptor signature for NP (Ciampi et al., CT-RSA '25)—and our implementation is an order of magnitude faster in overall signature generation, verification, and extraction.

Keywords: Adaptor Signatures · Online/Offline NIZKs · Post-Quantum Cryptography · VOLE-in-the-Head · Conditional Payments

* This is the full version of the paper accepted at Asiacrypt 2026.

Table of Contents

Practical Adaptor Signatures for NP from Online/Offline NIZK 1
Masayuki Abe, Dung Bui, Kelong Cong, Miyako Ohkubo, Zehua Shang, Akira Takahashi, and Mehdi Tibouchi
1 Introduction 3
1.1 Our Contributions 4
1.2 Applications 4
1.3 Related Work 5
2 Technical Overview 6
2.1 Generic Construction for Adaptor Signatures 6
2.2 Instantiation from VOLEitH-based Paradigm 8
3 Preliminaries 9
3.1 Adaptor Signatures 9
3.2 Non-Interactive Proof System 12
3.3 Digital Signature Scheme 13
4 Construction 13
4.1 Basic Scheme 13
4.2 Extension to Instance Hiding 16
5 VOLEitH-based Adaptor Signature 18
5.1 VOLE-in-the-Head Paradigm 18
5.2 Instantiation of ONIZK 19
6 Implementation and Experimental Results 23
A Related Notions 28
A.1 Security Notions for NIZKs 28
A.2 Definitions for Digital Signature Schemes 28
A.3 Other Security Notions for Adaptor Signatures 28
A.4 Variations of Instance Hiding Notions 30

1 Introduction

Digital signatures are a fundamental cryptographic primitive underpinning secure online payments over blockchain networks. In their most basic use case, a sender Alice digitally signs a transaction—e.g., "I agree to transfer x BTC to Bob"—and posts the signature on a public ledger, where it can be verified by anyone.

Motivating Application: Conditional Payments. A natural and practically important extension is to make payments *conditional*: for example, "I agree to transfer x BTC to Bob, provided he reveals to me a SHA-3 hash preimage y such that $H(y) = Y$", where Y is a publicly known hash value. A straightforward approach is to exploit the programmable scripting language of a platform such as Ethereum, where a smart contract can specify the condition and automatically release the funds upon its satisfaction. This naïve solution carries a serious privacy drawback: for the smart contract to verify the condition, the solution y must either be posted publicly on-chain, or at least be visible to the validators executing the contract. In many practical scenarios this is unacceptable, since y is meant to be communicated *privately* between the designated parties and should remain hidden from all other observers. One could attempt to restore privacy using zero-knowledge proofs (ZKPs): Alice attaches her public encryption key ek to the transaction, and Bob posts on-chain a ciphertext c together with a ZKP attesting that "c is an encryption of some y satisfying $R(Y, y) = 1$ under Alice's key ek." Upon verifying the proof, the smart contract releases the payment, and Alice decrypts c to recover y. While this approach is sound in principle, it inherently requires expensive proofs of plaintext knowledge and couples the payment flow to a public-key encryption scheme. A cleaner goal is to realize conditional payments entirely *without* encryption, keeping the on-chain footprint as lightweight as an ordinary digital signature.

Adaptor Signatures. Adaptor signatures, originally introduced by Poelstra [Poe17] under the name *scriptless scripts*, were developed precisely to address this limitation and to enable *off-chain* conditional payments. At a high level, a sender can generate a *pre-signature* $\tilde{\sigma}$ that is cryptographically bound to both a public instance Y and a transaction message, without yet constituting a valid signature. The pre-signature can be *adapted* into a complete, valid signature σ by any party who knows a witness y satisfying the relation $R(Y, y) = 1$. Crucially, the scheme satisfies *witness extractability*: given the pre-signature $\tilde{\sigma}$ and the resulting complete signature σ, the sender can efficiently extract a valid witness y. In a typical deployment, $\tilde{\sigma}$ is communicated out-of-band to a designated recipient, while the complete signature σ is posted publicly on-chain to trigger the coin transfer. Accordingly, adaptor signatures realize conditional payments between two designated parties without involving any trusted third-party checker and without compromising the privacy of the witness y. Beyond conditional payments, adaptor signatures find applications in atomic swaps [EFH+21], payment channel networks [MMK+17, MMS+19, MBB+19, AEE+21], oracle-based payment [MTV+23], coin-mixing [GMM+22, QPM+23], etc.

State of the Art and Limitations. As reviewed in Section 1.3, the majority of existing adaptor signature constructions support only a specific algebraic relation tied to the structure of the underlying signature scheme, such as the discrete-logarithm relation $Y = g^y$. While sufficient for simple transfer protocols, this restriction rules out the richer conditions (see Section 1.2), and motivates the study of adaptor signatures for *arbitrary* NP relations R. Dai et al. [DOY22] were the first to achieve this generality, but at the cost of witness privacy: their complete signature includes y in the clear. Liu et al. [LTZ24] addressed this by preserving privacy via a trapdoor-commitment framework; however, their only concrete instantiation is built on a ZKP for Hamiltonicity, so supporting any other NP relation requires a Karp reduction, rendering the approach highly impractical. Ciampi et al. [CLTZ25] recently proposed a practically-motivated approach that avoids Karp reductions entirely, building on MPC-in-the-head based verifiable encryption [TZ24]. They convert an MPC-in-the-head ZKP for the target NP relation into an adaptor signature by replacing the commitments to party views with public-key ciphertexts, enabling straight-line witness extraction without proofs of plaintext knowledge. While this is a significant step forward, the approach has two inherent limitations. First, the complete signature must carry one ciphertext per party view, causing the signature size to grow proportionally to the number of parties and repetitions in the protocol. Second, witness privacy relies on the IND-CPA security of the encryption scheme, introducing a computational assumption beyond what the underlying proof system requires. Both limitations are especially acute in the post-quantum setting. On the size front, even with the most compact post-quantum verifiable encryption [TZ24],

instantiating the AES-128 relation yields signatures of at least 28 KB (see Section 6), far too large for on-chain deployment. On the assumption front, witness privacy for this instantiation additionally requires Module-LWE. This state of affairs motivates the following central question:

Can we construct a practical, post-quantum adaptor signature for any NP relation under minimal computational assumptions?

1.1 Our Contributions

We make the following contributions towards practical post-quantum adaptor signatures for arbitrary NP relations. Our two main contributions are a generic construction framework and an efficient post-quantum instantiation thereof; we also contribute a refined security notion.

1. Generic Constructions for NP. We give two generic constructions of adaptor signature schemes for any NP relation R, built from a standard digital signature scheme and a new primitive we call *online/offline NIZK* (ONIZK). An ONIZK splits proof generation into an *offline phase*, executed before the witness y is known, and an efficient *online phase*, executed once y becomes available. We identify a key property, which we call *extraction with randomness*, stipulating that knowledge of the offline randomness r is sufficient to recover y from the online proof component. In our construction, the pre-signing algorithm runs the offline phase and signs the resulting component; adaptation runs the online phase using the witness; and witness extraction exploits extraction with randomness. Our *basic* construction achieves pre-signature adaptability, extractability, and witness hiding. Unlike [CLTZ25], witness hiding of our construction does not require any additional assumptions beyond those needed for ZK of ONIZK. This scheme is simple and efficient, and suitable for many applications of adaptor signatures that do not require stronger privacy guarantees. Our *instance-hiding* construction further guarantees instance hiding, a strong privacy notion introduced in this work (see below). To achieve this, the instance Y is encoded into an extended instance $(Y \oplus t_0, \mathsf{H}(t_0 \| t_1))$ using uniformly random masks t_0, t_1 and a hash function H. Provided that H is pseudorandom and collision resistant, as defined precisely later, the construction achieves instance hiding at the cost of a single additional hash evaluation in the relation circuit.

2. Refined Security Notions. To date, *witness hiding* [LTZ24] has been regarded as the minimal privacy requirement for adaptor signatures, while *unlinkability* [DOY22] represents the strongest known form of privacy. We revisit these notions and introduce a new notion called *instance hiding* (IH) that is both strictly stronger than unlinkability and admits a cleaner formulation. Informally, IH requires that an adversary holding the signing key *ask* cannot distinguish a complete signature obtained by adapting a pre-signature for instance Y from one generated directly by the signing algorithm; this captures a form of *forward security*, ensuring that complete signatures remain indistinguishable from ordinary ones on the same instance even after signing-key compromise. This guarantee is particularly valuable in settings such as atomic swaps, where Y is public and shared across multiple signatures.

In Table 1, we compare prior works with respect to witness- and instance-related privacy.

3. Efficient Post-Quantum Instantiation. We demonstrate the practicality of our framework by instantiating it within the *Vector Oblivious Linear Evaluation (VOLE) in the Head* paradigm for NP (VOLEitH for short) [BBD+23]. The offline/online structure of the VOLEitH protocol maps naturally to our ONIZK decomposition: the offline phase covers vector commitment setup and VOLE randomness generation, while the online phase executes the QuickSilver witness-binding proof, from which extraction with randomness follows directly. We instantiate the underlying signature scheme with FAEST (`https://faest.info/`), a compact AES-based post-quantum signature built on VOLEitH, yielding a post-quantum secure adaptor signature scheme for any NP relation. Compared to Ciampi et al. [CLTZ25], whose scheme requires the complete signature to carry $\tau \cdot N$ public-key ciphertexts (where τ is the number of repetitions and N is the number of parties) to enable straight-line witness extraction, our approach eliminates this ciphertext overhead entirely by relying instead on the offline randomness for extraction. For 128-bit security with the AES-128 relation, this reduces the complete signature size from 28 kB to 11.6 kB (see Section 6 and Table 2) and our overall signature generation, verification, and extraction are substantially faster.

1.2 Applications

Here we briefly review some applications of adaptor signatures for general NP relations other than post-quantum variants of the existing applications of adaptor signatures for specific relations.

Scheme	Inst. Hiding	Unlinkability	Wit. Hiding	Post-Quantum	Generic for NP
TMM21 [TMSM21]	-	-	-	✓	✗
EFHMR21 [EFH+21]	-	-	-	✗	✗
DOY22 [DOY22]	-	✓	-	✓	✓
LTZ24 [LTZ24]	-	-	✓	✓	✓
GSST24 [GSST24]	-	✓	-	✗	✗
CLTZ25 [CLTZ25]	-	-	✓	✓	✓
Ours	✓*	✓†	✓	✓	✓

Table 1. Comparison of security notions across adaptor signature schemes. We compare our construction with prior works satisfying the standard adaptor signature notions: pre-signature adaptability, extractability, and unforgeability. *Instance hiding is a new notion introduced in this work. †Unlinkability follows from instance hiding. "-" means the notion is not achieved or not considered in the paper. We note that the security definitions differ across the papers, details are discussed in Section 3 and Section A.3.

Risk-free Contingent Payments. When a seller Bob wants to sell a digital good y that satisfies a condition described as relation R on instance Y to a buyer Alice, they can engage in *zero-knowledge contingent payment (zkCP)* protocols [Bow16, CGGN17, Fuc19, NAA20] to achieve a fair exchange: Bob commits to y and proves to Alice its correctness in zero-knowledge. Alice then pays Bob, and Bob reveals y to Alice. Adaptor signatures can be used to realize zkCP in a more efficient and elegant way: Alice issues a pre-signature on relation R for Y to Bob, and Bob uses y to adapt it into a complete signature published on chain, which Alice can verify and extract y from. This way, Bob can sell y to Alice without trusting her to pay after receiving y, and Alice can buy y without trusting Bob to send it after receiving payment. Since the digital good y can be anything satisfying the relation R, this application is very general and can be used in various scenarios, such as selling credentials, decryption keys, or signatures on another message. As a concrete example requiring support for NP, suppose Alice wants to pay Bob in exchange for a valid, non-expired credential granting access to a service. Here, the witness $y = (\mathsf{sig}, \mathsf{msg}_{\mathsf{priv}})$ consists of a signature sig and a credential message $\mathsf{msg}_{\mathsf{priv}}$ encoding attributes such as an expiration date and access rights. The statement $Y = (\mathsf{vk}, \mathsf{msg}_{\mathsf{pub}})$ consists of the issuer's public verification key vk and a public component of the message $\mathsf{msg}_{\mathsf{pub}}$ (e.g., the service identifier). The relation $\mathcal{R}$ then checks that (i) sig is a valid signature under vk over $\mathsf{msg}_{\mathsf{priv}}||\mathsf{msg}_{\mathsf{pub}}$, (ii) the credential has not expired, and (iii) the attributes encoded in $\mathsf{msg}_{\mathsf{priv}}$ satisfy the required access policies.

One-Way Atomic Swaps. When Alice and Bob want to exchange signatures for authorizing transactions on different blockchains, adaptor signatures can realize an atomic swap [MMS+19, MMK+17]. Existing protocols typically require multiple rounds of interaction and/or multiple on-chain transactions. In a standard approach, Alice and Bob exchange pre-signatures on a common instance Y for relation R for which Alice knows a witness y. After Alice adapts Bob's pre-signature with y and posts the resulting complete signature on Alice's blockchain, Bob verifies it, extracts y, adapts Alice's pre-signature, and posts the complete signature on Bob's blockchain. This achieves atomicity, but it requires *exchanging* pre-signatures and it demands *script support on both blockchains* to verify them.

With adaptor signatures for general NP relations, Alice can instead send Bob a single pre-signature on relation R where Y encodes Bob's public key and the message describing the transaction on Bob's blockchain, and where R checks that the witness is a valid signature under Bob's public key for that message. Bob adapts the pre-signature into a complete signature and posts it on his blockchain to execute the transaction, while Alice extracts Bob's signature from the complete signature and posts it on her blockchain to execute the corresponding transaction there. This protocol only requires *one* pre-signature from Alice to Bob and allows *Alice's blockchain to remain scriptless*, since Bob's signature can be verified by the standard signature verification algorithm of Alice's blockchain.

1.3 Related Work

Other than the works on general NP relations mentioned above, there is a rich line of work on adaptor signatures for specific relations, which we briefly review here.

Discrete-logarithm setting. The original construction by Poelstra [Poe17] is built on Schnorr signatures and supports the discrete-logarithm relation $R_{\mathsf{DL}} = \{(Y, y) : Y = g^y\}$. Malavolta et al. [MMS+19] extended this line of work and gave a construction based ECDSA signatures, again for the discrete-logarithm relation. Aumayr et al. [AEE+21] provided the first formal security framework for adaptor signatures and prove the constructions based on Schnorr and ECDSA satisfy their security definition. Both constructions crucially exploit the additive homomorphism of the exponent: the complete signature is an ordinary Schnorr (resp. ECDSA) signature, so no on-chain overhead is incurred.

Post-quantum settings. Esgin et al. [EEE20] gave the first post-quantum adaptor signature, based on module lattices and supporting the Module-SIS relation $R_{\mathsf{SIS}} = \{(Y, y) : Y = \mathbf{A}\mathbf{y} \bmod q,\ \|\mathbf{y}\|_\infty \leq 1\}$. Tairi et al. [TMSM21] constructed an adaptor signature in the isogeny setting, supporting the group-action relation $R_{\mathsf{iso}} = \{(Y, y) : Y = [y]E_0\}$, where $[y]E_0$ denotes the action of a group element g^y on a base curve E_0. Meshram and Banerjee [MB26] gave a post-quantum construction based on the multivariate quadratic (MQ) assumption, supporting the hash relation $R_H = \{(Y, y) : Y = H(y)\}$; however, the witness y appears in plain in the complete signature, so witness privacy is not achieved. In all three post-quantum constructions, the supported relation is either dictated by the algebraic structure of the underlying signature or the construction sacrifices witness privacy.

Generic constructions for one-way homomorphisms. Erwig et al. [EFH+21] gave a generic construction from any *canonical identification scheme*, supporting any relation of the form $R_f = \{(Y, y) : Y = f(y)\}$ where f is the one-way group homomorphism implicit in the commitment phase of the identification protocol. Gerhart et al. [GSST24] extended this line of work to obtain constructions from Camenisch-Lysyanskaya, BBS+, and Waters signatures, again for relations of the form $Y = f(y)$ determined by the homomorphism built into each scheme.

Offline-Online NIZKs and Extractability. Delayed-input proof systems [CPS+16a, CPS+16b, COSV17, CPV20] are a special case of our offline-online NIZK formulation. They require the *instance* to be given with delay, i.e., only to the online algorithm, whereas ours only require the *witness* to be given with delay. Accordingly, their offline proof algorithm *must* operate without an instance, while ours *may* take the instance as input. This more general treatment permits broader instantiations.

Straight-line extractable proofs [GOS06, Gro06, Kat21, CDG+25] allow anyone holding the CRS trapdoor to extract a witness from *any* proof.[7] Our EwR property *restricts* extraction to proofs whose offline randomness is known to the extractor, enabling finer-grained and more efficient extraction suited to adaptor signature applications.

It is well known that knowledge of the ephemeral randomness in a public-coin proof such as Schnorr [Sch93] suffices to recover the witness. EwR formalizes this mechanism for offline-online NIZKs, tailored to the adaptor signature setting where the signer knows the offline randomness and later uses it to extract the witness. This contrasts with standard discrete-logarithm-based adaptor signatures [EFH+21, AEE+21, MMS+19, Poe17], where extraction works by subtracting the pre-signature $z = r + cx$ from the adapted signature $z' = y + r + cx$, requiring no knowledge of r.

2 Technical Overview

2.1 Generic Construction for Adaptor Signatures

An adaptor signature scheme AS for relation R is a tuple of polynomial-time algorithms $(\mathsf{Gen}, \mathsf{pSig}, \mathsf{pVer}, \mathsf{Adapt}, \mathsf{Ver}, \mathsf{Sig}, \mathsf{Ext})$ satisfying three standard security properties: *pre-signature adaptability*, *extractability* and *witness hiding*. We propose a generic construction realizing adaptor signatures for all NP languages. Let (Y, y) denote a statement–witness pair for R, i.e., $R(Y, y) = 1$. Our construction relies on two main building blocks: a variant of non-interactive zero-knowledge proofs (NIZKs) called *online/offline NIZK*, and a standard digital signature scheme. We first describe the online/offline NIZK and its required security properties, then present the adaptor signature construction.

Online/Offline NIZK. The online/offline paradigm splits proof generation into two phases, $\mathsf{P} := (\mathsf{P}_{\mathsf{off}}, \mathsf{P}_{\mathsf{on}})$. The *offline* phase $\mathsf{P}_{\mathsf{off}}$ can be executed independently of the witness: it takes randomness r as input and produces an offline proof component π_{off} deterministically. The *online* phase P_{on} is executed once the witness y becomes available and runs efficiently relative to the offline preprocessing.

[7] Straight-line extraction is also considered in the random oracle model, e.g., [Fis05, DFMS22, RT25], but the extractor does not work in real protocol executions, where the adversary's oracle queries are not observable.

Beyond the standard properties of soundness and zero-knowledge, we require an additional property called *extraction with randomness (EwR)*. Intuitively, the offline component π_{off} acts as a commitment to the randomness r, which is used to mask the witness y inside the online component π_{on}. Consequently, knowledge of r enables one to invert this masking and recover y. Formally, ONIZK satisfies EwR if there exists an efficient extractor EwR such that, for any (possibly maliciously crafted) online component π_{on}, if the full proof $\pi = (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})$ is valid and π_{off} was derived from r, then $\mathsf{EwR}(r, \pi)$ outputs a valid witness y for R. For instance, Schnorr proofs can be seen as a simple example of ONIZK with EwR, where the offline component π_{off} is the commitment $A = g^r$, and the online proof π_{on} is the response $s = r + c \cdot w$ computed deterministically with challenge c derived from the Fiat-Shamir transformation. Given r and s, one can extract witness w by computing $w = (s - r)/c$. A more involved example is MPC-in-the-Head proofs with preprocessing, such as the 5-round Katz–Kolesnikov–Wang (KKW) protocol [KKW18]: here, π_{off} consists of commitments to correlated randomness shares produced during an offline MPC execution, and these shares are used to mask the witness in the online phase.

Adaptor Signature Construction. Given an online/offline NIZK (ONIZK) $\mathsf{P} := (\mathsf{P}_{\mathsf{off}}, \mathsf{P}_{\mathsf{on}})$ and a standard signature $\mathsf{SIG} := (\mathsf{SIG.Gen}, \mathsf{Sig}, \mathsf{Ver})$. The AS construction is described as follows while the formal construction is shown in Figure 2:

- Key generation $\mathsf{AS.Gen}$: Runs $\mathsf{SIG.Gen}$ and $\mathsf{ONIZK.Setup}$ to produce a public/secret key pair (apk, ask), where $apk := (\mathsf{vk}, crs)$ and $ask := (\mathsf{sk}, crs)$. crs is for setup of ONIZK and $(\mathsf{vk}, \mathsf{sk})$ is a verification/signing key pair of signature.
- Pre-signing $\mathsf{AS.pSig}$: Given the secret key ask and a statement Y, the signer samples fresh randomness r and computes an offline NIZK component $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y; r)$. It then produces an underlying signature $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y \| \pi_{\mathsf{off}} \| m)$. The resulting pre-signature $\tilde{\sigma} := (s, r)$ binds the signer to the statement Y and π_{off}.
- Pre-signature verification $\mathsf{AS.pVer}$: Given a pre-signature $\tilde{\sigma} = (s, r)$ and statement Y, recomputes $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y; r)$ and checks the underlying signature via $\mathsf{SIG.Ver}$. Returns 1 if and only if the signature is valid over the expected message $Y \| \pi_{\mathsf{off}} \| m$.
- Adaptation $\mathsf{AS.Adapt}$: Given a pre-signature $\tilde{\sigma} = (s, r)$ and a witness y for the statement Y, computes the online NIZK component $\pi_{\mathsf{on}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{on}}(crs, Y, y, r)$ and combine them to the full proof $\pi := (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})$. The output is the full signature $\sigma := (Y, s, \pi)$. This is the step that "completes" the pre-signature once a valid witness becomes available.
- Full-signature verification $\mathsf{AS.Ver}$: Verify two checks jointly: it verifies the underlying signature over $Y \| \pi_{\mathsf{off}} \| m$ via $\mathsf{SIG.Ver}$, and verifies the NIZK proof π via $\mathsf{ONIZK.V}$. The algorithm returns 1 if and only if both checks pass.
- Direct signing $\mathsf{AS.Sig}$. Sampling a fresh statement–witness pair $(Y, y) \leftarrow \mathsf{Sample}_{\mathcal{LW}_R}(1^\lambda)$, generating the full NIZK proof $\pi \leftarrow \mathsf{ONIZK.P}(crs, Y, y)$, and producing the signature directly. This algorithm is used to generate signatures that are computationally indistinguishable from adapted ones, which will be essential for proving instance hiding.
- Witness extraction $\mathsf{AS.Ext}$: It is the inverse of $\mathsf{AS.Adapt}$. Given a pre-signature $\tilde{\sigma} = (s, r)$ and the corresponding full signature $\sigma = (Y, s', \pi)$, the extractor runs $\mathsf{ONIZK.EwR}(crs, \pi, r)$ to recover the witness y. Extraction succeeds because the randomness r was used to mask y inside π_{on} during adaptation, knowledge of r thus allows one to invert this masking and extract y.

Security of the Construction. The correctness of the proposed construction follows directly from the correctness of SIG and the completeness of ONIZK. We now turn to the security analysis.

- *Pre-signature Adaptability.* Follows directly from completeness of ONIZK. Given a valid pre-signature and a legitimate witness, $\mathsf{AS.Adapt}$ recomputes the offline component identically to $\mathsf{AS.pVer}$ and runs the online phase to complete the proof, which ONIZK completeness guarantees to be accepted.
- *Extractability.* Suppose an adversary produces a valid full signature σ^* on a fresh message m^* such that no valid witness can be extracted from any pre-signature in the query to pSig oracle. The security is reduced to EUF-CMA of Sig and EwR of ONIZK via a case split: either the adversary forges a fresh signature against SIG, or a matching pre-signature exists in the query OpSig, whose randomness is passed to recover the witness. Failure to extract in the latter case directly contradicts the EwR property.

– *Witness Hiding.* Reduced to adaptive zero-knowledge of ONIZK. The simulator replaces honest proof generation with a simulated proof that requires no knowledge of the witness, and since all other signature components are generated identically, any distinguishing advantage against witness hiding breaks the zero-knowledge property of ONIZK.

Extension to Instance Hiding. We extend our standard construction to get a variant which achieves a strong privacy notion, called *Instance Hiding*. Instance hiding is a new security notion that allows to hide the instance Y from adapted signature given adversary knows secret key *ask*. To support instance hiding, we encode the instance Y into an extended instance $Y_{\mathsf{ex}} := (Y \oplus t_0, \mathsf{H}(t_0||t_1))$ for a uniformly random value t_0 and t_1. Variable t_0 can be seen as a mask to hide Y and $\mathsf{H}(t_0||t_1)$ can be seen as a statistically hiding and computationally binding commitment to t_0. The extended witness relation R_{ex} is then defined over pairs $(Y_{\mathsf{ex}}, y_{\mathsf{ex}})$, where $Y_{\mathsf{ex}} = (C, h)$ and $y_{\mathsf{ex}} = (y, t_0, t_1)$, such that $R_{\mathsf{ex}}(Y_{\mathsf{ex}}, y_{\mathsf{ex}}) = 1$ if and only if $R(C \oplus t_0,\ y) = 1$ and $\mathsf{H}(t_0||t_1) = h$.

2.2 Instantiation from VOLEitH-based Paradigm

To demonstrate the efficiency of our construction, we instantiate it within the *VOLE-in-the-Head* (VOLEitH) paradigm. Concretely, the signature scheme Sign is instantiated using VOLEitH-based signature (FAEST), a post-quantum signature scheme supporting all NP languages. For the ONIZK instantiation, we carefully adapt the VOLEitH paradigm to the interactive setting by splitting it into two components with the help of a random oracle, corresponding to the offline phase $\mathsf{ONIZK.P}_{\mathsf{off}}$ and the online phase $\mathsf{ONIZK.P}_{\mathsf{on}}$, respectively, as shown in Figure 1.

The VOLEitH paradigm [BBD+23] constructs a public-coin zero-knowledge proof for any NP relation by combining three main steps. First, an all-but-one vector commitment scheme $(\mathsf{VC.P}, \mathsf{VC.Open}, \mathsf{VC.V})$ commits to a set of random seeds, which are then used to generate a batch of small VOLE correlations via a divide-and-conquer approach [Roy22]. Second, these correlations are merged into a big one over $\mathbb{F}_{2^\lambda}$ using $\mathsf{VOLE.P}$ and verified through a derandomization and consistency check step using $(\mathsf{VOLE.C}, \mathsf{VOLE.V})$. Finally, the resulting VOLE correlation is fed into the QuickSilver protocol [YSWW21] $(\mathsf{P_{QS}}, \mathsf{V_{QS}})$, a VOLE-based ZKP that proves circuit satisfiability gate by gate using IT-MAC authenticated values. We refer the reader to Section 5.1 for the details of notations and construction.

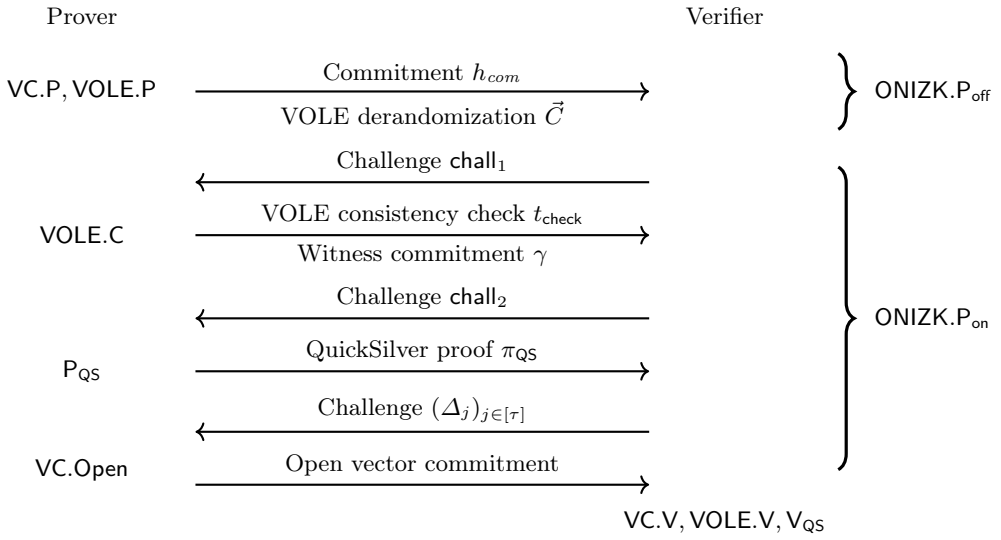

Fig. 1. ONIZK instantiation from VOLEitH Paradigm

Our construction exploits a key property of the QuickSilver (QS) proof: the witness is not needed to define the offline phase, which consists solely of setting up the vector commitment and generating the VOLE randomness. The online phase, in contrast, includes constructing the VOLE correlation and executing the QS proof. Extractability with randomness follows from this separation: the witness

commitment γ in QS proof, sent by the prover during the online phase, masks the witness using randomness that is only determined offline. Consequently, an extractor can recover the witness if and only if it has access to this offline randomness.

3 Preliminaries

Notation. For a randomized algorithm Alg, we write $y \leftarrow \mathsf{Alg}(x; r)$, where x is the input supplied by the caller and r denotes the algorithm's randomness. If r has not been specified previously in the surrounding context, it is sampled uniformly from Alg's randomness space and made explicit to the srrounding context so that it can be reused in a later invocation of Alg, or given to another party. If it has been specified explicitly, that value of r is supplied to Alg as its randomness. We leave the randomness space of r implicit, as it depends on the actual instantiation and is unimportant throughout the paper. This dual use is clear from context.

Let R be a polynomial-time computable predicate that $R(Y, y)$ runs in polynomial time in $|Y|$ and outputs 0 or 1. We say that R is a witness relation for language $\mathcal{L}_R \in \mathsf{NP}$ if $\mathcal{L}_R = \{Y \in \{0,1\}^* : \exists y \in \{0,1\}^{\mathrm{poly}(|Y|)} \text{ s.t. } R(Y, y) = 1\}$ for a fixed polynomial poly. In this case, we also say that y is a witness for Y with respect to R. Let $\mathcal{LW}_R := \{(Y, y) : Y \in \mathcal{L}_R \wedge R(Y, y) = 1\}$ be the set of instance-witness pairs. Let $\overline{\mathcal{L}}_R = \{0,1\}^* \setminus \mathcal{L}_R$ be the set of instances not in $\mathcal{L}_R$. By $\mathcal{L}_R(\lambda)$, $\mathcal{LW}_R(\lambda)$, and $\overline{\mathcal{L}}_R(\lambda)$, we denote the sets of instances, instance-witness pairs, and non-instances with size restricted to λ, respectively. Hence, $\mathcal{L}_R = \cup_{\lambda\in\mathbb{N}}\mathcal{L}_R(\lambda)$, $\mathcal{LW}_R = \cup_{\lambda\in\mathbb{N}}\mathcal{LW}_R(\lambda)$, and $\overline{\mathcal{L}}_R = \cup_{\lambda\in\mathbb{N}}\overline{\mathcal{L}}_R(\lambda)$.

By $\mathsf{Sample}_{\mathcal{L}_R}(\lambda)$, we denote a p.p.t. algorithm that samples Y from $\mathcal{L}_R(\lambda)$ according to some distribution. Similarly by $\mathsf{Sample}_{\mathcal{LW}_R}$ we denote the same as $\mathsf{Sample}_{\mathcal{L}_R}$ that additionally outputs a witness.

Definition 1 (Hard Language/Relation). *A language $\mathcal{L}$ (and a witness relation R associated to it) is hard if there exists a p.p.t. algorithm $\mathsf{Sample}_{\mathcal{L}_R}$ that, for any p.p.t. machine $\mathcal{A}$, the following advantage is negligible in λ:*

$$\mathsf{Adv}^{\mathsf{hl}}_{\mathsf{Sample}_{\mathcal{L}_R},\mathcal{A}}(\lambda) := \Pr[R(Y, \mathcal{A}(Y)) = 1 : Y \leftarrow \mathsf{Sample}_{\mathcal{L}_R}(\lambda)]$$

If $\mathcal{L}_R$ is hard, it holds that sampling from $\mathcal{L}_R(\lambda)$ and $\mathcal{LW}_R(\lambda)$ polynomial number of times yields two identical instances happens only with negligible probability.

3.1 Adaptor Signatures

We follow the syntactical definition from [LTZ24].

Definition 2 (Adaptor Signature Scheme). *An adaptor signature scheme AS for relation R is a tuple of polynomial-time algorithms, $(\mathsf{Gen}, \mathsf{pSig}, \mathsf{pVer}, \mathsf{Adapt}, \mathsf{Ver}, \mathsf{Sig}, \mathsf{Ext})$ that:*

$\mathsf{Gen}(1^\lambda) \to (apk, ask)$ *: is a probabilistic algorithm that takes security parameter λ and outputs a public and private key pair (apk, ask).*

$\mathsf{pSig}(ask, Y, m) \to \tilde{\sigma}$ *: is a probabilistic algorithm that takes ask, instance Y, and message m, as input and outputs a pre-signature $\tilde{\sigma}$.*

$\mathsf{pVer}(apk, m, Y, \tilde{\sigma}) \to 0/1$ *: is a deterministic algorithm that takes apk, m, Y, $\tilde{\sigma}$, and outputs 0 or 1 for rejection and acceptance, respectively.*

$\mathsf{Adapt}(apk, \tilde{\sigma}, Y, y, m) \to \sigma$ *: takes apk, $\tilde{\sigma}$, Y, witness y, and m as input, and outputs signature σ.*

$\mathsf{Ver}(apk, m, \sigma) \to 0/1$ *: is a deterministic algorithm that takes apk, σ, and m as input, and outputs 0 or 1 for rejection and acceptance, respectively.*

$\mathsf{Sig}(ask, m) \to \sigma$ *: is a probabilistic algorithm that takes ask and m as input and generates a signature σ.*

$\mathsf{Ext}(apk, \tilde{\sigma}, \sigma) \to y$ *: is a deterministic algorithm that takes apk, $\tilde{\sigma}$, and σ as input, and outputs witness y.*

The algorithms must satisfy the correctness conditions: For any large enough $\lambda > 0$, $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$, $m \in \{0,1\}^*$, $(Y, y) \leftarrow \mathcal{L}_R(\lambda)$, and $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$, it holds that:

- *(Pre-signature correctness)* $\mathsf{pVer}(apk, m, Y, \tilde{\sigma}) = 1$,

– *(Adapted Signature correctness)* $\mathsf{Ver}(apk, m, \mathsf{Adapt}(apk, \tilde{\sigma}, Y, y, m)) = 1$,
– *(Signature correctness)* $\mathsf{Ver}(apk, m, \mathsf{Sig}(ask, m)) = 1$, *and*
– *(Extraction correctness)* $R(Y, \mathsf{Ext}(apk, \tilde{\sigma}, \mathsf{Adapt}(apk, m, \tilde{\sigma}, y))) = 1$.

If no pre-signatures are observed, an adaptor signature scheme can be considered as a regular signature scheme and it should satisfy standard notion of unforgeability (EUF-CMA). On the other hand, if a pre-signature is given on an instance selected by the adversary, it is trivially possible to adapt the pre-signature using a valid witness for the instance. Therefore, the security definition of adaptor signatures should capture the idea that adapted signatures on fresh messages (i.e., messages never signed by the legitimate signer) cannot be produced without a corresponding pre-signature generated by the legitimate signer and a valid witness for the embedded instance. There are several formal definitions for capturing this idea, and we follow the notion of *Extractability* from [DOY22] in Definition 3 below.

Definition 3 (Extractability [DOY22]). *An adaptor signature scheme* AS *(for relation R) is extractable, if for every p.p.t. adversary $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{ext}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \Pr[\mathrm{Expr}^{\mathsf{ext}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1]$$

is negligible in λ where $\mathrm{Expr}^{\mathsf{ext}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ is defined as follows.

$\underline{\mathrm{Expr}^{\mathsf{ext}}_{\mathsf{AS},\mathcal{A}}(\lambda)}$
1: $Q_{\mathsf{pSig}} := \emptyset$
2: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$
3: $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OpSig}(\cdot,\cdot)}(apk)$
4: $b_1 := (\mathsf{Ver}(apk, m^*, \sigma^*) = 1)$
5: $b_2 := (\forall (Y, \tilde{\sigma}) \in Q_{\mathsf{pSig}}[m^*] \;\; R(Y, \mathsf{Ext}(apk, \tilde{\sigma}, \sigma^*, Y)) \neq 1)$
6: **return** $b_1 \wedge b_2$

$\underline{\mathsf{OpSig}(m, Y)}$
1: $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$
2: $Q_{\mathsf{pSig}}[m] := Q_{\mathsf{pSig}}[m] \cup \{(Y, \tilde{\sigma})\}$
3: **return** $\tilde{\sigma}$

Pre-signature adaptability ensures that any pre-signature passing verification can be correctly adapted using a legitimate witness. The notion is often defined with a perfect or statistical flavor, e.g., [LTZ24, GSST24]. A computational variant is defined in [DOY22], which we follow with a correction that the adversary is given *ask* instead of *apk*.

Definition 4 (Pre-signature Adaptability [DOY22]). *An adaptor signature scheme* AS *for relation R is pre-signature adaptable, if, for any p.p.t. adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{psa}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \Pr[\mathrm{Expr}^{\mathsf{psa}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1]$ is negligible in λ, where $\mathrm{Expr}^{\mathsf{psa}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ is:*

$\underline{\mathrm{Expr}^{\mathsf{psa}}_{\mathsf{AS},\mathcal{A}}(\lambda)}$
1: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(\tilde{\sigma}^*, Y^*, y^*, m^*) \leftarrow \mathcal{A}(ask)$
3: $\sigma^* \leftarrow \mathsf{Adapt}(apk, \tilde{\sigma}^*, Y^*, y^*, m^*)$
4: $b_1 := \mathsf{pVer}(apk, m^*, Y^*, \tilde{\sigma}^*)$
5: $b_2 := R(Y^*, y^*)$
6: $b_3 := \mathsf{Ver}(apk, m^*, \sigma^*)$
7: **return** $b_1 \wedge b_2 \wedge \neg b_3$

Witness hiding property of adaptor signatures captures the idea that adapted signatures do not leak information about the witness. It is implemented with a simulation-based definition where a simulator creates adapted signatures without knowing the witness. Since the notion in the previous works has minor issues (see discussion in Section A.3), we present a refined, stronger definition below. (It is stronger in the sense that the adversary is given *ask* instead of the signing oracles. It captures the case where the signer is breached.)

We also introduce the notion of *instance hiding*, which hides the instance Y from adapted signatures. It is defined similarly to witness hiding except that the signature is generated without accessing to the specified witness. We therefore present the definitions together in the following.

Definition 5 (Witness/Instance Hiding). *An adaptor signature scheme* AS *for relation* R *is witness hiding (*aWH*), if there exists a pair of p.p.t. algorithms* $\mathsf{Sim} := (\mathsf{SimGen}, \mathsf{SimSig})$ *that, for any p.p.t. stateful adversary* $\mathcal{A}$*, advantage*

$$\mathsf{Adv}^{\mathsf{awh}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{Expr}^{\mathsf{awh\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{awh\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda) \to 1]\right|$$

is negligible in λ*, where* $\mathrm{Expr}^{\mathsf{awh\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ *and* $\mathrm{Expr}^{\mathsf{awh\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)$ *are defined as follows.*

$\underline{\mathrm{Expr}^{\mathsf{awh\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)}$
1: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(Y, y, m, \mathsf{state}) \leftarrow \mathcal{A}(ask)$
3: $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$
4: $\sigma \leftarrow \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)$
5: $b_1 := \mathcal{A}(\mathsf{state}, \sigma)$
6: $b_2 := (R(Y, y) = 1)$
7: **return** $b_1 \wedge b_2$

$\underline{\mathrm{Expr}^{\mathsf{awh\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)}$
1: $(apk, ask, \tau) \leftarrow \mathsf{SimGen}(1^\lambda)$ ▷
$(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$ for IH
2: $(Y, y, m, \mathsf{state}) \leftarrow \mathcal{A}(ask)$
3: $\sigma \leftarrow \mathsf{SimSig}(ask, m, Y, \tau)$ ▷
$\sigma \leftarrow \mathsf{Sig}(ask, m)$ for IH
4: $b_1 := \mathcal{A}(\mathsf{state}, \sigma)$
5: $b_2 := (R(Y, y) = 1)$
6: **return** $b_1 \wedge b_2$

An adaptor signature scheme AS *for relation* R *is instance hiding (*IH*) if, for any p.p.t. stateful adversary* $\mathcal{A}$*, advantage*

$$\mathsf{Adv}^{\mathsf{ih}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1]\right|$$

is negligible in λ*, where* $\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ *is exactly the same as* $\mathrm{Expr}^{\mathsf{awh\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$*, and* $\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ *is defined by replacing simulator* $(\mathsf{SimGen}, \mathsf{SimSig})$ *in* $\mathrm{Expr}^{\mathsf{awh\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)$ *with real algorithms* $(\mathsf{Gen}, \mathsf{Sig})$*.*

We stress that the above notion of IH compares an adapted signature to a signature generated by Sig, rather than to a simulated one. This choice enables the implication to the notion of *unlinkability* [DOY22] defined in Definition 6, a useful property in applications to privacy-preserving rollups as claimed in [DOY22]. We present more variations of IH and discuss their implications and separations in Section A.4.

Definition 6 (Unlinkability). *Adaptor signature scheme* AS *is unlinkable (*UL*) if, for any p.p.t. adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{ul}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{Expr}^{\mathsf{ul},0}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{ul},1}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1]\right|$$

is negligible in λ*, where* $\mathrm{Expr}^{\mathsf{ul},\mathsf{b}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ *for* $b \in \{0, 1\}$ *are defined as follows.*

$\underline{\mathrm{Expr}^{\mathsf{ul},\mathsf{b}}_{\mathsf{AS},\mathcal{A}}(\lambda)}$
1: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $\tilde{b} \leftarrow \mathcal{A}^{\mathsf{Sig}(ask,\cdot),\mathsf{pSig}(ask,\cdot,\cdot),\mathsf{O}_b(\cdot,\cdot,\cdot)}(apk)$
3: ***return*** $\tilde{b}$

$\underline{\mathsf{O}_0(m, Y, y)}$
1: $\mathsf{Assert}(R(Y, y) = 1)$
2: $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$
3: $\sigma \leftarrow \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)$
4: ***return*** σ

$\underline{\mathsf{O}_1(m, Y, y)}$
1: $\mathsf{Assert}(R(Y, y) = 1)$
2: $\sigma \leftarrow \mathsf{Sig}(ask, m)$
3: ***return*** σ

Theorem 1 ($\mathsf{IH} \Rightarrow \mathsf{UL}$)**.** *If an adaptor signature scheme* AS *for relation* R *is instance hiding, then it is unlinkable.*

Proof. We construct an instance hiding adversary $\mathcal{A}$ given black-box access to an unlinkability adversary $\mathcal{B}$. A key observation is that $\mathcal{A}$ is given ask in the instance hiding game, and can simulate the pre-signing and signing oracles for $\mathcal{B}$ using ask. Single access to O_b in the unlinkability game is simulated by using a signature given in the instance hiding game. Multiple access to O_b is addressed by standard hybrid argument. Detailed game transitions follow.

Let q denote the number of queries from $\mathcal{B}$ to O_b. We define sequence of games Game 0 to Game q as follows.

– Game 0: This is $\mathrm{Expr}^{\mathsf{ul},0}_{\mathsf{AS},\mathcal{B}}(\lambda)$ where adversary $\mathcal{B}$ plays the unlinkability game with O_0.
– Game i: The game is modified in a way that the j-th query to oracle O_b is answered by O_0 if $i < j$ and by O_1 if $i \geq j$.
– Game q: This is $\mathrm{Expr}^{\mathsf{ul},1}_{\mathsf{AS},\mathcal{B}}(\lambda)$ where adversary $\mathcal{B}$ plays the unlinkability game with O_1.

Let P_i denote the probability that $\mathcal{B}$ outputs 1 in Game i. By definition, $|P_0 - P_q| = \mathsf{Adv}^{\mathsf{ul}}_{\mathsf{AS},\mathcal{B}}(\lambda)$.

We claim that $P_i - P_{i+1} \leq \mathsf{Adv}^{\mathsf{ih}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ for adversary $\mathcal{A}$. To prove the claim, we construct $\mathcal{A}$ playing the instance hiding game as follows. Given ask, $\mathcal{A}$ runs $\mathcal{B}$ with apk as input (it is implicit that apk is available from ask). For all queries from $\mathcal{B}$ to $\mathsf{Sig}(ask, \cdot)$, $\mathsf{pSig}(ask, \cdot, \cdot)$, $\mathcal{A}$ simulates the oracles generating signatures and pre-signatures with ask, and forwarding the signatures to $\mathcal{B}$. For j-th query to O_b with input (m, Y, y) from $\mathcal{B}$, $\mathcal{A}$ does the following:

$i < j$: $\mathcal{A}$ simulates O_0 with ask, and sends the resultings σ to $\mathcal{B}$.
$i = j$: $\mathcal{A}$ outputs (Y, y, m) and receives σ. It then forwards σ to $\mathcal{B}$.
$i > j$: $\mathcal{A}$ simulates O_1 with ask, and sends the resultings σ to $\mathcal{B}$.

When $\mathcal{B}$ outptus $\tilde{b}$, $\mathcal{A}$ outputs $\tilde{b}$ as well. This completes the description of $\mathcal{A}$.

Now, if $\mathcal{A}$ is playing $\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$, the signature σ it received is an adapted one and the view of $\mathcal{B}$ is the same as that of Game i. Otherwise, if $\mathcal{A}$ is playing $\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda)$, the signature σ it received is the one created from Sig, and the view of $\mathcal{B}$ is the same as that of Game $i+1$. Thus, $P_i - P_{i+1} = \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] \leq \mathsf{Adv}^{\mathsf{ih}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ as claimed.

By summing up, we have $\mathsf{Adv}^{\mathsf{ul}}_{\mathsf{AS},\mathcal{B}}(\lambda) = |P_0 - P_q| = |(P_0 - P_1) + (P_1 - P_2) + \cdots + (P_{q-1} - P_q)| \leq q \cdot \mathsf{Adv}^{\mathsf{ih}}_{\mathsf{AS},\mathcal{A}}(\lambda)$. Thus, if $\mathsf{Adv}^{\mathsf{ih}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ is negligible, so is $\mathsf{Adv}^{\mathsf{ul}}_{\mathsf{AS},\mathcal{B}}(\lambda)$. □

3.2 Non-Interactive Proof System

A non-interactive proof system consists of a tuple of algorithms, $\mathsf{NIZK} := (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ that:

– $\mathsf{Setup}(1^\lambda) \to crs$: is a probabilistic algorithm that takes security parameter λ and outputs a common reference string crs.
– $\mathsf{P}(crs, Y, y) \to \pi$: is a probabilistic algorithm, called a prover algorithm, that takes crs, instance Y, and witness y as input, and outputs a proof π.
– $\mathsf{V}(crs, Y, \pi) \to 0/1$: is a deterministic algorithm, called a verifier algorithm, that takes crs, Y, and π as input, and outputs a bit representing rejection or acceptance.

In the random oracle model, the setup algorithm Setup and crs can be empty, i.e., the random oracle itself is considered as a result of the setup. Algorithms P and V are given access to the random oracle. This is the case for our instantiation in Section 5 where Setup and crs are dropped from the description.

We follow the standard notion of completeness, soundness, and adaptive zero-knowledge for NIZK as presented in Appendix A.1.

For our construction of adaptor signatures, we require the underlying NIZK to have two additional properties: offline/online proof generation and extraction with randomness. The offline/online property separates the proof generation into two parts, $\mathsf{P} := (\mathsf{P}_{\mathsf{off}}, \mathsf{P}_{\mathsf{on}})$,where the offline part $\mathsf{P}_{\mathsf{off}}$ can be computed before the witness is known, and the online part P_{on} can be computed efficiently once the witness is available. Such a structure is often employed in efficient NIZKs for NP, especially when the offline part involves heavy computations that can be pre-computed.

Definition 7 (Offline/Online NIZK). *A non-interactive proof system is online/offline if proof algorithm* P *is split into two parts,* $\mathsf{P} := (\mathsf{P}_{\mathsf{off}}, \mathsf{P}_{\mathsf{on}})$*, such that* $\mathsf{P}_{\mathsf{off}}(crs, Y; r) \to \pi_{\mathsf{off}}$*, and* $\mathsf{P}_{\mathsf{on}}(crs, Y, y, r) \to \pi_{\mathsf{on}}$ *and the resulting proof is* $\pi := (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})$*.*

We exploit the offline/online structure of NIZK not only for the efficiency but also for the novel insight that the offline proof π_{off} can be seen as a commitment of randomness r, which is often used to hide the witness y in the online proof π_{on}. In short, this property requires that if r is known, one can reliably extract y from any valid π_{on}. This is a common property observed in many NIZKs such as Schnorr [Sch93] and MPC-in-the-Head proofs with preprocessing [KKW18]. It is also crucial for our adaptor signature construction. We formalize this property as follows.

Definition 8 (Extraction with Randomness: EwR). *An online/offline NIZK for relation R is extractable with randomness (EwR) if there exists a deterministic polynomial-time algorithm* EwR *that, for all sufficiently large λ, $crs \leftarrow \mathsf{Setup}(1^\lambda)$, $(Y, y) \in \mathcal{LW}_R$, $\pi_{\mathsf{off}} \leftarrow \mathsf{P}_{\mathsf{off}}(crs, Y; r)$, $\pi_{\mathsf{on}} \leftarrow \mathsf{P}_{\mathsf{on}}(crs, Y, y, r)$, it holds that $R(Y, \mathsf{EwR}(crs, Y, \pi, r)) = 1$. In addition, it is required that, for any p.p.t. adversary $\mathcal{A}$, probability*

$$\mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l|l} \mathsf{V}(crs, Y, \pi) = 1 \wedge & crs \leftarrow \mathsf{Setup}(1^\lambda) \\ R(Y, y) \neq 1 & (Y, \mathsf{state}) \leftarrow \mathcal{A}(crs) \\ & \pi_{\mathsf{off}} \leftarrow \mathsf{P}_{\mathsf{off}}(crs, Y; r) \\ & \pi_{\mathsf{on}} \leftarrow \mathcal{A}(\mathsf{state}, r) \\ & \pi := (\pi_{\mathsf{off}}, \pi_{\mathsf{on}}) \\ & y \leftarrow \mathsf{EwR}(crs, \pi, r) \end{array}\right]$$

is negligible in λ.

Definition 9 (Adaptive Zero-Knowledge). *A non-interactive proof system for relation R is adaptive zero-knowledge if there exists p.p.t. algorithms* $\mathsf{SimSetup}$ *and* SimP *such that for all p.p.t. adversary $\mathcal{A}$, advantage*

$$\mathsf{Adv}^{\mathsf{azk}}_{\mathit{NIZK},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{Expr}^{\mathsf{azk},0}_{\mathit{NIZK},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{azk},1}_{\mathit{NIZK},\mathcal{A}}(\lambda) \to 1]\right|$$

is negligible in λ, where $\mathrm{Expr}^{\mathsf{azk},b}_{\mathit{NIZK},\mathcal{A}}(\lambda)$ for $b \in \{0, 1\}$ are defined as follows.

$\underline{\mathrm{Expr}^{\mathsf{azk},0}_{\mathsf{NIZK},\mathcal{A}}(\lambda)}$
1: $crs \leftarrow \mathsf{Setup}(1^\lambda)$
2: **return** $\mathcal{A}^{\mathsf{O}_0(\cdot,\cdot)}(crs)$

$\underline{\mathsf{O}_0(Y, y)}$
1: **assert** $R(Y, y) = 1$
2: $\pi \leftarrow \mathsf{P}(crs, Y, y)$
3: **return** π

$\underline{\mathrm{Expr}^{\mathsf{azk},1}_{\mathsf{NIZK},\mathcal{A}}(\lambda)}$
1: $(crs, \tau) \leftarrow \mathsf{SimSetup}(1^\lambda)$
2: **return** $\mathcal{A}^{\mathsf{O}_1(\cdot,\cdot)}(crs)$

$\underline{\mathsf{O}_1(Y, y)}$
1: **assert** $R(Y, y) = 1$
2: $\pi \leftarrow \mathsf{SimP}(\tau, Y)$
3: **return** π

3.3 Digital Signature Scheme

We follow the standard syntactical and security definitions for digital signature scheme (see Appendix A.2 for formal definitions). A digital signature scheme, SIG, is a tuple of algorithms $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ that satisfy the correctness condition. It is existential unforgeability against chosen message attacks (EUF-CMA security) if any p.p.t. adversary with access to a signing oracle can only forge a signature on a fresh message with negligible probability. By $\mathsf{Adv}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$, we denote the advantage function for adversary $\mathcal{A}$ winning the EUF-CMA game for signature scheme SIG.

4 Construction

4.1 Basic Scheme

The basic adaptor signature scheme is built from a standard digital signature scheme SIG and an online/offline NIZK proof system ONIZK. Let $\mathsf{ONIZK} = (\mathsf{Setup}, \mathsf{P}_{\mathsf{off}}, \mathsf{P}_{\mathsf{on}}, \mathsf{V}, \mathsf{EwR})$ be an online/offline NIZK proof system that is EwR. Let $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a digital signature scheme. At a high level, the pre-signing algorithm runs the offline phase $\mathsf{ONIZK.P}_{\mathsf{off}}$ on a fresh randomness r to produce an offline proof component π_{off}, and signs the string $Y \| \pi_{\mathsf{off}} \| m$ under $\mathsf{SIG.Sig}$, outputting $\tilde{\sigma} = (s, r)$. Adaptor algorithm then runs the online phase $\mathsf{ONIZK.P}_{\mathsf{on}}$ using the witness y to complete the proof, while witness extraction exploits the fact that knowledge of r is sufficient to recompute y inside π_{on} via $\mathsf{ONIZK.EwR}$. Figure 2 shows the construction of basic adaptor signature scheme AS for R from ONIZK and SIG.

Theorem 2 (Correctness). *The adaptor signature scheme in Figure 2 is correct as in Definition 2 if* SIG *is correct and* ONIZK *is complete.*

$\underline{(apk, ask) \leftarrow \mathsf{AS.Gen}(1^\lambda)}$

1: $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$
2: $crs \leftarrow \mathsf{ONIZK.Setup}(1^\lambda)$
3: $apk := (\mathsf{vk}, crs),\ \ ask := (\mathsf{sk}, crs)$
4: **return** (apk, ask)

$\underline{\tilde{\sigma} \leftarrow \mathsf{AS.pSig}(ask, Y, m)}$

1: $(\mathsf{sk}, crs) \leftarrow ask$
2: $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y; r)$
3: $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y \| \pi_{\mathsf{off}} \| m)$
4: **return** $\tilde{\sigma} := (s, r)$

$\underline{0/1 \leftarrow \mathsf{AS.pVer}(apk, \tilde{\sigma}, Y, m)}$

1: $(\mathsf{vk}, crs) \leftarrow apk,\ (s, r) \leftarrow \tilde{\sigma}$
2: $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y; r)$
3: $b \leftarrow \mathsf{SIG.Ver}(\mathsf{vk}, s, Y \| \pi_{\mathsf{off}} \| m)$
4: **return** b

$\underline{\sigma \leftarrow \mathsf{AS.Adapt}(apk, \tilde{\sigma}, Y, y, m)}$

1: $(\mathsf{vk}, crs) \leftarrow apk,\ (s, r) \leftarrow \tilde{\sigma}$
2: $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y; r)$
3: $\pi_{\mathsf{on}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{on}}(crs, Y, y, r)$
4: $\pi := (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})$
5: **return** $\sigma := (Y, s, \pi)$

$\underline{0/1 \leftarrow \mathsf{AS.Ver}(apk, \sigma, m)}$

1: $(\mathsf{vk}, crs) \leftarrow apk,\ (Y, s, \pi) \leftarrow \sigma$
2: $(\pi_{\mathsf{off}}, \pi_{\mathsf{on}}) \leftarrow \pi$
3: $b_1 \leftarrow \mathsf{SIG.Ver}(\mathsf{vk}, s, Y \| \pi_{\mathsf{off}} \| m)$
4: $b_2 \leftarrow \mathsf{ONIZK.V}(crs, Y, \pi)$
5: **return** $b_1 \wedge b_2$

$\underline{\sigma \leftarrow \mathsf{AS.Sig}(ask, m)}$

1: $(\mathsf{sk}, crs) \leftarrow ask$
2: $(Y, y) \leftarrow \mathsf{Sample}_{\mathcal{LW}_R}(1^\lambda)$
3: $\pi \leftarrow \mathsf{ONIZK.P}(crs, Y, y)$
4: $(\pi_{\mathsf{off}}, \pi_{\mathsf{on}}) \leftarrow \pi$
5: $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y \| \pi_{\mathsf{off}} \| m)$
6: **return** $\sigma := (Y, s, \pi)$

$\underline{y \leftarrow \mathsf{AS.Ext}(apk, \tilde{\sigma}, \sigma)}$

1: $(\mathsf{vk}, crs) \leftarrow apk,\ (s, r) \leftarrow \tilde{\sigma}$
2: $(Y, s', \pi) \leftarrow \sigma$
3: $y \leftarrow \mathsf{ONIZK.EwR}(crs, \pi, r)$
4: **return** y

Fig. 2. Basic adaptor signature scheme for relation R.

The above can be proved by inspecting the construction. We omit the details but note that it relies on the perfect correctness of SIG and ONIZK, as well as perfect correctness of the extractor EwR for the latter. If we are to allow negligible errors in relevant components, it results in negligible correctness error for the adaptor signature scheme as well.

Theorem 3 (Pre-signature Adaptability). *The adaptor signature scheme in Figure 2 is pre-signature adaptable as in Definition 4 if* ONIZK *is complete.*

Proof. First observe that the CRS crs is honestly generated by ONIZK.Setup in Gen. Suppose that $\mathcal{A}$ outputs $(\tilde{\sigma}^*, Y^*, y^*, m^*)$ where $\tilde{\sigma}^* = (s^*, r^*)$ is a valid pre-signature for m^* and Y^*, and $R(Y^*, y^*) = 1$. Observe that pVer verifies s^* with respect to a message including $\pi^*_{\mathsf{off}} = \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y^*; r^*)$. Adapt generates π^*_{off} in the same way, and then computes $\pi^*_{\mathsf{on}} \leftarrow \mathsf{ONIZK.P}(crs, Y^*, y^*)$. Since $(\pi^*_{\mathsf{off}}, \pi^*_{\mathsf{on}})$ is created following the legitimate procedure with Y^* and y^* satisfying $R(Y^*, y^*) = 1$ and ONIZK is complete, ONIZK.V in Ver accepts the proof. Since s^* is a valid signature for $Y^* || \pi^*_{\mathsf{off}} || m^*$, SIG.Ver also accepts. Hence Ver accepts the adapted signature. □

The above assumes perfect completeness of ONIZK. If we allow negligible completeness error, then we have to assume that it is hard to find r such that $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y^*; r)$ and $\pi_{\mathsf{on}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{on}}(crs, Y^*, y^*)$ satisfy $\mathsf{ONIZK.V}(crs, Y^*, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})) = 0$. In the random oracle model, this can be achieved by hashing r and using the output for the offline proof.

Theorem 4 (Extractability). *The adaptor signature scheme in Figure 2 is extractable as in Definition 3 if* SIG *is* EUF-CMA *and* ONIZK *is* EwR.

Proof. Let $(apk, ask) = ((\mathsf{vk}, crs), (\mathsf{sk}, crs))$ be the key pair from $\mathsf{AS.Gen}(1^\lambda)$ in $\mathrm{Expr}^{\mathsf{ext}}_{\mathsf{AS},\mathcal{A}}(\lambda)$, and let (m^*, σ^*) with $\sigma^* = (Y^*, s^*, (\pi^*_{\mathsf{off}}, \pi^*_{\mathsf{on}}))$ be the output of $\mathcal{A}$. Suppose that $\mathrm{Expr}^{\mathsf{ext}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ outputs 1. Then, $\mathsf{SIG.Ver}(\mathsf{vk}, s^*, Y^* || \pi^*_{\mathsf{off}} || m^*) = 1$, $\mathsf{ONIZK.V}(crs, Y^*, (\pi^*_{\mathsf{off}}, \pi^*_{\mathsf{on}})) = 1$, and $R(Y, \mathsf{AS.Ext}(apk, \tilde{\sigma}, \sigma^*)) \neq 1$ holds for every $(Y, \tilde{\sigma}) \in Q_{\mathsf{pSig}}[m^*]$.

Define event S that there exists $(Y^*, \tilde{\sigma}^*) \in Q_{\mathsf{pSig}}[m^*]$ with $\tilde{\sigma}^* = (s, r^*)$ such that $\mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y^*; r^*) = \pi^*_{\mathsf{off}}$. In other words, the oracle OpSig signed the exact string $Y^* || \pi^*_{\mathsf{off}} || m^*$ with the underlying scheme SIG. By $\overline{\mathsf{S}}$ we denote that event S did not occur.

Case $\overline{\mathsf{S}}$. We claim that $\Pr[\mathcal{A}\text{ wins} \wedge \overline{\mathsf{S}}] \leq \mathsf{Adv}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{B}}(\lambda)$ for some adversary $\mathcal{B}$. First observe that when $\overline{\mathsf{S}}$ holds, the string $M^* := Y^*||\pi^*_{\mathsf{off}}||m^*$ was never signed with SIG through OpSig. Hence, (s^*, M^*) is a fresh valid forgery for SIG.

We then construct $\mathcal{B}$ against EUF-CMA security of SIG as follows. Given vk, $\mathcal{B}$ runs $crs \leftarrow \mathsf{ONIZK.Setup}(1^\lambda)$, sets $apk := (\mathsf{vk}, crs)$ and calls $\mathcal{A}(apk)$. It answers $\mathsf{OpSig}(m, Y)$ by choosing fresh r, computing $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y; r)$, querying its CMA signing oracle on $Y||\pi_{\mathsf{off}}||m$ to get s, and returning (s, r). On the final output of $\mathcal{A}$, $\mathcal{B}$ outputs (M^*, s^*). As already observed, whenever $\mathcal{A}$ wins with $\overline{\mathsf{S}}$, this constitutes a valid EUF-CMA forgery, so

$$\Pr[\mathcal{A}\text{ wins} \wedge \overline{\mathsf{S}}] \leq \mathsf{Adv}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{B}}(\lambda).$$

Case S. We claim $\Pr[\mathcal{A}\text{ wins} \wedge \mathsf{S}] \leq q_{\mathsf{OpSig}} \cdot \mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{C}}(\lambda)$ for some adversary $\mathcal{C}$. Intuition is the following. Let $\tilde{\sigma}^* = (s, r^*)$ be the matching pre-signature in $Q_{\mathsf{pSig}}[m^*]$. By construction, $\mathsf{AS.Ext}(apk, \tilde{\sigma}^*, \sigma^*) = \mathsf{ONIZK.EwR}(crs, (\pi^*_{\mathsf{off}}, \pi^*_{\mathsf{on}}), r^*)$. Since $\mathcal{A}$ wins, $\mathsf{ONIZK.V}(crs, Y^*, (\pi^*_{\mathsf{off}}, \pi^*_{\mathsf{on}})) = 1$ holds but $R(Y^*, \mathsf{ONIZK.EwR}(crs, (\pi^*_{\mathsf{off}}, \pi^*_{\mathsf{on}}), r^*)) \neq 1$. This is exactly a violation of the EwR property of ONIZK.

We now construct $\mathcal{C}$ against the EwR property of ONIZK as follows. Given crs, $\mathcal{C}$ runs $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$, sets $apk := (\mathsf{vk}, crs)$, $ask := (\mathsf{sk}, crs)$, and guesses an index $i^* \in [q_{\mathsf{OpSig}}]$ uniformly at random. It invokes $\mathcal{A}(apk)$ and answers all queries to OpSig honestly using ask, except for the i^*-th query (m^*, Y^*). At that point, $\mathcal{C}$ outputs Y^* to its ewr challenger (the first call of the adversary in the ewr game), receiving in return the randomness r^* used to compute $\pi^*_{\mathsf{off}} = \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y^*; r^*)$. $\mathcal{C}$ then computes $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y^*||\pi^*_{\mathsf{off}}||m^*)$, returns $\tilde{\sigma}^* := (s, r^*)$ to $\mathcal{A}$, and continues the simulation. When $\mathcal{A}$ halts with output (m^*, σ^*), $\mathcal{C}$ outputs π^*_{on} as the online proof (Phase 2 of the ewr game).

The simulation is perfect, since r^* is uniformly random and all other answers are computed honestly. Whenever $\mathcal{A}$ wins with S and the guess i^* matches the target pre-signature, $\mathcal{C}$ wins its ewr game. The probability that the guess is correct is $1/q_{\mathsf{OpSig}}$, so

$$\Pr[\mathcal{A}\text{ wins} \wedge \mathsf{S}] \leq q_{\mathsf{OpSig}} \cdot \mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{C}}(\lambda).$$

Combining both cases, we obtain

$$\mathsf{Adv}^{\mathsf{we}}_{\mathsf{AS},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{B}}(\lambda) + q_{\mathsf{OpSig}} \cdot \mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{C}}(\lambda).$$

This is negligible whenever SIG is EUF-CMA and ONIZK is EwR. □

Theorem 5 (Witness Hiding). *The adaptor signature scheme in Figure 2 is witness hiding as in Definition 5 if* ONIZK *is adaptive zero-knowledge.*

Proof. We construct $\mathsf{Sim} = (\mathsf{SimGen}, \mathsf{SimSig})$ as follows.

$\underline{(apk, ask, \tau) \leftarrow \mathsf{SimGen}(1^\lambda)}$
1: $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$
2: $(crs, \tau) \leftarrow \mathsf{ONIZK.SimSetup}(1^\lambda)$
3: $apk := (\mathsf{vk}, crs)$, $ask := (\mathsf{sk}, crs)$
4: **return** (apk, ask, τ)

$\underline{\sigma \leftarrow \mathsf{SimSig}(ask, m, Y, \tau)}$
1: $(\mathsf{sk}, crs) \leftarrow ask$
2: $(\pi_{\mathsf{off}}, \pi_{\mathsf{on}}) \leftarrow \mathsf{ONIZK.SimP}(\tau, Y)$
3: $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y\|\pi_{\mathsf{off}}\|m)$
4: **return** $\sigma := (Y, s, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}}))$

With respect to the above Sim, the difference between two experiments $\mathsf{Expr}^{\mathsf{awh\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ and $\mathsf{Expr}^{\mathsf{awh\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)$ is that in the former, proof $(\pi_{\mathsf{off}}, \pi_{\mathsf{on}})$ is generated honestly by $\mathsf{ONIZK.P}_{\mathsf{off}}$ and $\mathsf{ONIZK.P}_{\mathsf{on}}$, while in the latter it is generated by the zero-knowledge simulator $(\mathsf{ONIZK.SimSetup}, \mathsf{ONIZK.SimP})$. All other components of the signature are generated in the same way in both experiments. We thus have a perfect reduction from the advantage of $\mathcal{A}$ in distinguishing $\mathsf{Expr}^{\mathsf{awh\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ and $\mathsf{Expr}^{\mathsf{awh\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)$ to the advantage of a constructed adversary $\mathcal{B}$ in breaking the zero-knowledge property of ONIZK. We thus have

$$\mathsf{Adv}^{\mathsf{awh}}_{\mathsf{AS},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{azk}}_{\mathsf{ONIZK},\mathcal{B}}(\lambda),$$

which is negligible whenever ONIZK is adaptive zero-knowledge. □

4.2 Extension to Instance Hiding

We introduce *instance hiding*, a stronger privacy notion formally defined in Definition 5. We compare it with witness hiding and prove that instance hiding implies unlinkability in Theorem 1. Full details, including the discussion about the notion, are presented in Section 3 alongside the necessary background for ease of reference.

To support instance hiding, we consider encoding Y into an extended instance $Y_{\mathsf{ex}} := (Y \oplus t_0, \mathsf{H}(t_0||t_1))$ where t_0 and t_1 are fresh randomnesses, and H is a hash function. The witness relation R_{ex} is defined as the relation between an extended instance $Y_{\mathsf{ex}} = (C, h)$ and an extended witness $y_{\mathsf{ex}} = (y, t_0, t_1)$ such that $R_{\mathsf{ex}}(Y_{\mathsf{ex}}, y_{\mathsf{ex}}) = 1$ if and only if $R(C \oplus t_0, y) = 1$ and $\mathsf{H}(t_0||t_1) = h$.

Let ONIZK be an online/offline NIZK proof system for relation R_{ex} that is extractable with randomness. We construct an instance-hiding adaptor signature scheme AS for R from ONIZK and SIG, as shown in Figure 3. The construction follows the basic scheme in Figure 2, except that both the instance and witness are encoded into an extended form, and the NIZK is used to prove the extended relation $\{(y, t_0, t_1) : R(C \oplus t_0) = 1 \wedge \mathsf{H}(t_0||t_1) = h\}$. The security proof is also nearly identical to the one for the basic construction, except that we rely on H being pseudo-random to argue that the adversary learns no information about Y from Y_{ex}, and that the extractor can still recover a valid witness for Y from the extracted witness for Y_{ex}. Analogously to $\mathsf{Sample}_{\mathcal{LW}_R}$, we write $\mathcal{L}_{\mathsf{ex}}(\lambda)$ for the sampler that draws $(Y, y) \leftarrow \mathsf{Sample}_{\mathcal{LW}_R}(\lambda)$, samples fresh $t_0||t_1 \leftarrow \{0,1\}^{|Y|+\lambda}$, and outputs $(Y_{\mathsf{ex}}, y_{\mathsf{ex}}) := ((Y \oplus t_0, \mathsf{H}(t_0||t_1)), (y, t_0, t_1))$; this is the sampler used by $\mathsf{AS.Sig}$ in Figure 3.

$\underline{(apk, ask) \leftarrow \mathsf{AS.Gen}(1^\lambda)}$
1: $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$
2: $crs \leftarrow \mathsf{ONIZK.Setup}(1^\lambda)$
3: $apk := (\mathsf{vk}, crs)$
4: $ask := (\mathsf{sk}, crs)$
5: **return** (apk, ask)

$\underline{\tilde{\sigma} \leftarrow \mathsf{AS.pSig}(ask, Y, m)}$
1: $(\mathsf{sk}, crs) \leftarrow ask$
2: $t_0||t_1 \leftarrow \{0,1\}^{|Y|+\lambda}$
3: $Y_{\mathsf{ex}} := (Y \oplus t_0, \mathsf{H}(t_0||t_1))$
4: $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y_{\mathsf{ex}}; r)$
5: $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y_{\mathsf{ex}}\|\pi_{\mathsf{off}}\|m)$
6: **return** $\tilde{\sigma} := (s, r, t_0, t_1)$

$\underline{0/1 \leftarrow \mathsf{AS.pVer}(apk, \tilde{\sigma}, Y, m)}$
1: $(\mathsf{vk}, crs) \leftarrow apk$
2: $(s, r, t_0, t_1) \leftarrow \tilde{\sigma}$
3: $Y_{\mathsf{ex}} := (Y \oplus t_0, \mathsf{H}(t_0||t_1))$
4: $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y_{\mathsf{ex}}; r)$
5: $b \leftarrow \mathsf{SIG.Ver}(\mathsf{vk}, s, Y_{\mathsf{ex}}\|\pi_{\mathsf{off}}\|m)$
6: **return** b

$\underline{\sigma \leftarrow \mathsf{AS.Adapt}(apk, \tilde{\sigma}, Y, y, m)}$
1: $(\mathsf{vk}, crs) \leftarrow apk$
2: $(s, r, t_0, t_1) \leftarrow \tilde{\sigma}$
3: $Y_{\mathsf{ex}} := (Y \oplus t_0, \mathsf{H}(t_0||t_1))$
4: $y_{\mathsf{ex}} := (y, t_0, t_1)$
5: $\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(crs, Y_{\mathsf{ex}}; r)$
6: $\pi_{\mathsf{on}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{on}}(crs, Y_{\mathsf{ex}}, y_{\mathsf{ex}}, r)$
7: **return** $\sigma := (Y_{\mathsf{ex}}, s, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}}))$

$\underline{0/1 \leftarrow \mathsf{AS.Ver}(apk, \sigma, m)}$
1: $(\mathsf{vk}, crs) \leftarrow apk$
2: $(Y_{\mathsf{ex}}, s, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})) \leftarrow \sigma$
3: $b_1 \leftarrow \mathsf{SIG.Ver}(\mathsf{vk}, s, Y_{\mathsf{ex}}\|\pi_{\mathsf{off}}\|m)$
4: $b_2 \leftarrow \mathsf{ONIZK.V}(crs, Y_{\mathsf{ex}}, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}}))$
5: **return** $b_1 \wedge b_2$

$\underline{\sigma \leftarrow \mathsf{AS.Sig}(ask, m)}$
1: $(\mathsf{sk}, crs) \leftarrow ask$
2: $(Y_{\mathsf{ex}}, y_{\mathsf{ex}}) \leftarrow \mathcal{L}_{\mathsf{ex}}(\lambda)$
3: $(\pi_{\mathsf{off}}, \pi_{\mathsf{on}}) \leftarrow \mathsf{ONIZK.P}(crs, Y_{\mathsf{ex}}, y_{\mathsf{ex}})$
4: $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y_{\mathsf{ex}}\|\pi_{\mathsf{off}}\|m)$
5: **return** $\sigma := (Y_{\mathsf{ex}}, s, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}}))$

$\underline{y \leftarrow \mathsf{AS.Ext}(apk, \tilde{\sigma}, \sigma)}$
1: $(\mathsf{vk}, crs) \leftarrow apk$
2: $(s, r, t_0, t_1) \leftarrow \tilde{\sigma}$
3: $(Y_{\mathsf{ex}}, s', (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})) \leftarrow \sigma$
4: $y_{\mathsf{ex}} \leftarrow \mathsf{ONIZK.EwR}(crs, Y_{\mathsf{ex}}, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}}), r)$
5: $(y, t_0', t_1') \leftarrow y_{\mathsf{ex}}$
6: **return** y

Fig. 3. Adaptor signature scheme with instance hiding property. ONIZK proves $\{(y, t_0, t_1) : R(C \oplus t_0, y) = 1 \wedge \mathsf{H}(t_0||t_1) = h\}$ for $Y_{\mathsf{ex}} = (C, h)$.

Below, we prove the security properties of AS in Figure 3. Correctness is by inspection. We focus on Instance Hiding, pre-signature adaptability, and Extractability.

Theorem 6 (Instance Hiding). *If* ONIZK *is adaptive zero-knowledge and* H *is pseudo-random function satisfying Definition 10, then the adaptor signature scheme* AS *in Figure 3 is instance hiding.*

Definition 10 (Pseudo-randomness of H). $\mathsf{H} : \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{\mathrm{poly}(\lambda)}$ *is pseudo-random if, for any p.p.t. algorithm* $\mathcal{A}$*, advantage*

$$\mathsf{Adv}^{\mathsf{prg}}_{\mathsf{H},\mathcal{A}}(\lambda) := \left| \Pr_{t_0,t_1}[\mathcal{A}(t_0, \mathsf{H}(t_0||t_1)) \to 1] - \Pr_{t_0,h}[\mathcal{A}(t_0, h) \to 1] \right|$$

is negligible in λ.

Proof. The proof is by a sequence of hybrids where we gradually transform game $\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ into $\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda)$. In the first hybrid, we replace $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$ and $\sigma \leftarrow \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)$ in steps 3 and 4 of $\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ with $\mathsf{Sign}(ask, m)[Y, y]$ defined as follows:

$\underline{\sigma \leftarrow \mathsf{Sign}(ask, m)[Y, y]}$
1: $(\mathsf{sk}, crs) \leftarrow ask$
2: $t_0||t_1 \leftarrow \{0,1\}^{|Y|+\lambda}$
3: $Y_{\mathsf{ex}} := (Y \oplus t_0, \mathsf{H}(t_0||t_1))$
4: $y_{\mathsf{ex}} := (y, t_0, t_1)$
5: $(\pi_{\mathsf{off}}, \pi_{\mathsf{on}}) \leftarrow \mathsf{ONIZK.P}(crs, Y_{\mathsf{ex}}, y_{\mathsf{ex}})$
6: $s \leftarrow \mathsf{SIG.Sig}(\mathsf{sk}, Y_{\mathsf{ex}} \| \pi_{\mathsf{off}} \| m)$
7: **return** $\sigma := (Y_{\mathsf{ex}}, s, (\pi_{\mathsf{off}}, \pi_{\mathsf{on}}))$

Observe that the distribution of $\mathsf{Sign}(ask, m)[Y, y]$ is identical to that of $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$ and $\sigma \leftarrow \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)$ in the real game.

In the second hybrid, we modify the first hybrid by replacing the real NIZK proof with a simulated one. Precisely, $\mathsf{ONIZK.Setup}$ in Gen is replaced with $\mathsf{ONIZK.SimSetup}$ and $\mathsf{ONIZK.P}$ in $\mathsf{Sign}(ask, m)[Y, y]$ is replaced with $\mathsf{ONIZK.SimP}$. The distribution of crs and σ in the second hybrid is indistinguishable from that in the first hybrid by the zero-knowledge property of ONIZK. The advantage of the adversary is thus upper bounded by $\mathsf{Adv}^{\mathsf{azk}}_{\mathsf{NIZK},\mathcal{B}}(\lambda)$.

In the third hybrid, we modify $\mathsf{Sign}(ask, m)[Y, y]$ in the second hybrid by replacing Y_{ex} and y_{ex} computed from the given Y and y with a randomly generated Y_{ex} and y_{ex}. More precisely, we replace steps 2, 3, and 4 of $\mathsf{Sign}(ask, m)[Y, y]$ with $(Y_{\mathsf{ex}}, y_{\mathsf{ex}}) \leftarrow \mathcal{L}_{\mathsf{ex}}(\lambda)$. Observe that, at this point, y_{ex} (in particular t_1) is not used in the computation of σ and is independent of the view of the adversary. We justify the difference by constructing three sub-hybrids starting from the second hybrid and moving towards the third hybrid. In the first sub-hybrid, $\mathsf{H}(t_0||t_1)$ is replaced with a random h. Since H is pseudo-random satisfying Definition 10, this hybrid is bounded by $\mathsf{Adv}^{\mathsf{prg}}_{\mathsf{H},\mathcal{C}}(\lambda)$. We then make the second sub-hybrid that replaces $Y \oplus t_0$ with $Y' \oplus t'_0$ in Y_{ex} sampled uniformly from $\mathcal{L}_{\mathsf{ex}}(\lambda)$. Since t_0 is uniformly random and independent from all other parts of the adversary's view and $Y' \oplus t'_0$ is also uniformly random, this modification gives no advantage to the adversary. We then make the final sub-hybrid that replaces the random h with $\mathsf{H}(t'_0||t'_1)$ in Y_{ex} sampled from $\mathcal{L}_{\mathsf{ex}}(\lambda)$. As in the first sub-hybrid, the advantage from this hybrid is bounded by $\mathsf{Adv}^{\mathsf{prg}}_{\mathsf{H},\mathcal{C}}(\lambda)$. At this point, we have reached the third hybrid. In total, the advantage of the adversary is bounded by $2 \cdot \mathsf{Adv}^{\mathsf{prg}}_{\mathsf{H},\mathcal{C}}(\lambda)$.

In the fourth hybrid, we modify the third hybrid by reverting the second hybrid's zero-knowledge substitution: $\mathsf{ONIZK.SimSetup}$ in Gen is replaced with $\mathsf{ONIZK.Setup}$, and $\mathsf{ONIZK.SimP}$ in $\mathsf{Sign}(ask, m)[Y, y]$ is replaced with $\mathsf{ONIZK.P}$, now run on the randomly generated Y_{ex} and y_{ex}. By the zero-knowledge property of ONIZK, the distribution of crs and σ in the fourth hybrid is computationally indistinguishable from that in the third hybrid. The advantage of the adversary is thus upper bounded by $\mathsf{Adv}^{\mathsf{azk}}_{\mathsf{NIZK},\mathcal{B}}(\lambda)$.

Finally, observe that the distribution of σ in the fourth hybrid is identical to that in the $\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ since $\mathsf{Sign}(ask, m)[Y, y]$ in the fourth hybrid is identical to $\mathsf{Sign}(ask, m)$.

Accumulating the advantages of the adversary in all hybrids, we have that the advantage $\mathsf{Adv}^{\mathsf{ih}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ of the adversary in distinguishing $\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ and $\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ is upper bounded by $2 \cdot \mathsf{Adv}^{\mathsf{azk}}_{\mathsf{NIZK},\mathcal{B}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{prg}}_{\mathsf{H},\mathcal{C}}(\lambda)$. □

Theorem 7 (Pre-signature Adaptability). *The adaptor signature scheme in Figure 3 is pre-signature adaptable as in Definition 4 if* ONIZK *is complete.*

Proof. The argument is the same as in the proof of Theorem 3 for the basic scheme. The CRS is honestly generated by ONIZK.Setup in Gen, and from a valid pre-signature $\tilde{\sigma}^*$ we recover the underlying instance as $Y^* := Y^*_{\mathsf{ex}} \oplus t^*_0$. By completeness of ONIZK, this yields $R(Y^*, y^*) = 1$, so the remainder of the proof follows exactly as in Theorem 3. □

Theorem 8 (Extractability). *The adaptor signature scheme in Figure 3 is extractable as in Definition 3 if* SIG *is* EUF-CMA *and* ONIZK *is* EwR *and* H *is target-collision resistant.*

First of all, we define the target-collision resistance of a hash function H as follows.

Definition 11 (Target-Collision Resistance). *A hash function* $\mathsf{H} : \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{\mathrm{poly}(\lambda)}$ *is target-collision resistant if, for any p.p.t. algorithm* $\mathcal{A}$*, probability*

$$\mathsf{Adv}^{\mathsf{tcol}}_{\mathsf{H},\mathcal{A}}(\lambda) := \Pr_{t \leftarrow \{0,1\}^{\ell(\lambda)}}[t' \leftarrow \mathcal{A}(t) : t' \in \{0,1\}^{\ell(\lambda)} \wedge t' \neq t \wedge \mathsf{H}(t') = \mathsf{H}(t)]$$

is negligible in λ*.*

Proof. The proof follows the same structure as Theorem 4 for the basic scheme, with the only extra consideration that the adversary's forgery is for an extended instance $Y^*_{\mathsf{ex}} = (C^*, h^*)$. Define event S as in the proof of Theorem 4.

Case $\overline{\mathsf{S}}$. The reduction to EUF-CMA for SIG is identical to the basic case.

Case S. From a successful forgery we obtain a valid NIZK proof for Y^*_{ex} and run the extractor ONIZK.EwR. If the extracted witness is invalid for Y^*_{ex}, then we break the EwR property of ONIZK. Otherwise the extractor outputs an extended witness (y^*, t^*_0, t^*_1) with $\mathsf{H}(t^*_0||t^*_1) = h^*$. If $C^* \oplus t^*_0$ is not equal to Y^*, then we can construct adversary $\mathcal{D}$ that breaks the target collision resistance of H since t^*_0 is different from the original t_0 used in the pre-signature, but $\mathsf{H}(t^*_0||t^*_1) = \mathsf{H}(t_0||t_1)$.

As in Theorem 4, the reduction guesses the target pre-signature among q_{OpSig} queries, so

$$\Pr[\mathcal{A} \text{ wins} \wedge \mathsf{S}] \leq q_{\mathsf{OpSig}}(\mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{C}}(\lambda) + \mathsf{Adv}^{\mathsf{tcol}}_{\mathsf{H},\mathcal{D}}(\lambda)).$$

Combining both cases gives

$$\mathsf{Adv}^{\mathsf{we}}_{\mathsf{AS},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{B}}(\lambda) + q_{\mathsf{OpSig}}(\mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{C}}(\lambda) + \mathsf{Adv}^{\mathsf{tcol}}_{\mathsf{H},\mathcal{D}}(\lambda)).$$

This is negligible whenever SIG is EUF-CMA, ONIZK is EwR, and H is target-collision resistant. □

5 VOLEitH-based Adaptor Signature

In this section, we construct a post-quantum adaptor signature for NP language based on our ONIZK-based generic construction. Our adaptor signature is leveraged from the VOLE-in-the-Head (VOLEitH) paradigm. In Section 5.1, we briefly describe the framework in an interactive setting and its components to maintain the security. We show how to instantiate ONIZK by combining this framework and Fiat-Shamir heuristic in Section 5.2, and we then prove that our ONIZK satisfies the extraction with randomness property.

5.1 VOLE-in-the-Head Paradigm

We instantiate an offline/online NIZK for any NP relation R with $(Y, \vec{y}) \in R$[8], using the VOLEitH paradigm [BBD+23] which can be decomposed into three components as shown in Figure 4. The first component is an all-but-one vector commitment scheme used to commit to the prover's randomness. This committed randomness later allows the prover and verifier to generate the required VOLE

[8] In this section, we write the witness as a bit vector $\vec{y} \in \{0,1\}^k$, rather than y as elsewhere in the paper, since the VOLEitH paradigm operates directly on the bit-level representation of the witness (e.g., wire values of the predicate circuit $\mathcal{C}$).

correlations. Once the VOLE correlation is constructed, combining with a witness and a VOLE-based zero-knowledge proof system (QuickSilver) [YSWW21], we obtain a public-coin ZKP.

Formally, given that $Y, \vec{y} \in \{0,1\}^k$ are statement and witness, respectively, $\mathcal{C}$ is the predicate circuit of Y, i.e., $\mathcal{C}(\vec{y}) = 1$, below we briefly describe the main ingredients of the VOLEitH-based public-coin ZKP for any relation $(Y, \vec{y}) \in R$.

- Given $r \leftarrow_r \{0,1\}^\lambda, N = \mathsf{poly}(\lambda)$, all-but-one vector commitment to commit one-out-of-N random seeds $(\mathsf{sd}_i)_{i\in[N]}$. It consists of three PPT algorithms:
  $\mathsf{VC.Com}(r, N) \to (com, (\mathsf{sd}_i)_{i\in[N]})$: generate the random seeds and their commitment.
  $\mathsf{VC.Open}(\Delta \leftarrow_r [N], r) \to (open, \overline{com})$: generate the opening for all seeds except the Δ-th seed sd_Δ and compute a commitment for the hidden seed.
  $\mathsf{VC.V}(open, \Delta, com, \overline{com}) \to (b, (\mathsf{sd}_i)_{i\neq\Delta})$: from $open, \overline{com}$ recompute com and verify it(output 1 if it is valid) and reconstruct all-but-one seeds.
- A deterministic algorithm that uses outputs from the vector commitment scheme to build a random VOLE correlation over the field $\mathbb{F}_{2^\lambda}$. While the prover gets $(\vec{u}, \vec{v}) \in \mathbb{F}_2^\ell \times \mathbb{F}_{2^\lambda}^\ell$ and the verifier gets $(\Delta, \vec{w}) \in \mathbb{F}_{2^\lambda} \times \mathbb{F}_{2^\lambda}^\ell$. These values satisfy the relation: $\vec{w} = \Delta \cdot \vec{u} + \vec{v}$.
  To efficiently obtain a VOLE correlation over $\mathbb{F}_{2^\lambda}$, a divide-and-conquer method [Roy22] is applied. This method combines τ independent vector commitments, each of length N, where $N^\tau = 2^\lambda$. Each vector commitment produces its own VOLE correlation. To merge them into a single bigger correlation (over $\mathbb{F}_{2^\lambda}$) while keeping the same $\vec{u}$ in every instance, the prover sends a derandomization vector $\vec{C} \in \mathbb{F}_2^{(\tau-1)\ell}$ to the verifier.
  Finally, to ensure the combined correlation is executed correctly, a consistency check is performed. This check uses a challenge value chall_1 as input.
  $\mathsf{VOLE.P}((\mathsf{sd}_i^j)_{i\in[N],j\in[\tau]}, \ell) \to (\vec{u}, \vec{v}, \vec{C})$: prover produces the small VOLE correlations and the derandomization to get the big VOLE correlation.
  $\mathsf{VOLE.C}(\vec{u}, \vec{v}, \mathsf{chall}_1) \to t_{\mathsf{check}}$: the consistency check.
  $\mathsf{VOLE.V}(\mathsf{chall}_1, t_{\mathsf{check}}, \vec{C}, (\Delta_j, (\mathsf{sd}_i^j)_{i\neq\Delta_j})_{j\in[\tau]}, \ell) \to (b, \Delta, \vec{w})$, where $b = 1$ if $\vec{w} = \Delta \cdot \vec{u} + \vec{v}$ otherwise $b = 0$: verifier constructs VOLE correlation and checks the consistency. We note that $(\vec{C}, (\Delta_j, (\mathsf{sd}_i^j)_{i\neq\Delta_j})_{j\in[\tau]})$ is used to define and compute $(\Delta, \vec{w})$, while $(\mathsf{chall}_1, t_{\mathsf{check}})$ is used to check whether $\vec{w} = \Delta \cdot \vec{u} + \vec{v}$.
- A VOLE-based zero-knowledge proof (ZKP), specifically the QuickSilver protocol built on IT-MAC values $((\vec{u}, \vec{v}), (\Delta, \vec{w}))$, to prove circuit satisfiability gate by gate. The IT-MAC is used to authenticate all wire values in the circuit. We split the QuickSilver proof into two main steps: 1) Generating authenticated values for the witness, 2) Proving the correctness of multiplication gates.
  First, for the input wires (i.e., the witness $\vec{y}$), the prover commits to $\vec{y} \in \{0,1\}^k$ by derandomizing the VOLE correlation. Concretely, the prover sends $\gamma := \vec{y} - \vec{u}_{[0,k-1]}$, which allows the verifier to reconstruct authenticated values of $\vec{y}$.
  Second, multiplication gates correspond to degree-2 polynomial constraints. These are verified using the remaining part of the VOLE correlation. Formally, the proof is described as below given a challenge chall_2:

$$\mathsf{P}_{\mathsf{QS}}(\mathcal{C}, \vec{y}, \vec{u}, \vec{v}, \mathsf{chall}_2) \to \pi_{\mathsf{QS}}, \quad \mathsf{V}_{\mathsf{QS}}(Y, \Delta, \vec{w}, \gamma, \pi_{\mathsf{QS}}, \mathsf{chall}_2) \to 0/1.$$

5.2 Instantiation of ONIZK

The online/offline NIZK is built on top of VOLEitH proof system, combined with Fiat-Shamir transformation to remove interaction. The full construction is in Figure 5 and the formal VOLEitH-based ZK in interactive setting is shown in Figure 4.

Our construction takes advantage of the property that, during the execution of the QuickSilver (QS) proof, the witness is not required to define the offline phase. The online phase, in contrast, consists of constructing the VOLE correlation and executing the QS proof. Extractability is achieved by observing that the derandomized vector γ sent from the prover to the verifier contains information about the witness. Specifically, the witness can be extracted if and only if the extractor knows the randomness used to mask it, and this randomness is generated during the online phase. To prove the extraction with randomness property, we combine the binding of vector commitment, VOLE consistency check, QS soundness while assume that the collision of hash functions is negligible. Compare

VOLEitH Paradigm

Parameters

- $(Y, \vec{y}) \in R$, $\vec{y} \in \{0,1\}^k$, $\mathcal{C}(\vec{y}) = 1$
- System parameters: $\ell \in \mathbb{N}$, $N = 2^n$, $\tau = O(1)$, $N^\tau = 2^\lambda$
- Random oracle $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{2\lambda}$

Protocol

- P constructs vector commitment and VOLE derandomization:
  - $(r_j)_{j\in[\tau]} \leftarrow \{0,1\}^{\lambda\tau}$
  - For each $j \in [\tau]$: $(com_j, (\mathsf{sd}_i^j)_{i\in[N]}) \leftarrow \mathsf{VC.Com}(r_j, N)$
  - Compute $h_{com} := \mathsf{H}(com_1, \cdots, com_\tau)$
  - $(\vec{u}, \vec{v}, \vec{C}) \leftarrow \mathsf{VOLE.P}((\mathsf{sd}_i^j)_{i\in[N],j\in[\tau]}, \ell)$
  - Send $h_{com}, \vec{C}$ to V
- V sends a random challenge chall_1 to P
- P computes consistency check and witness commitment:
  - $t_{\mathsf{check}} \leftarrow \mathsf{VOLE.C}(\vec{u}, \vec{v}, \mathsf{chall}_1)$
  - $\gamma := \vec{y} - \vec{u}_{[0,k-1]}$
  - Send $\gamma, t_{\mathsf{check}}$ to V
- V sends a random challenge chall_2 to P
- P runs the QuickSilver proof: $\pi_{\mathsf{QS}} \leftarrow \mathsf{P}_{\mathsf{QS}}(\mathcal{C}, \vec{y}, \vec{u}, \vec{v}, \mathsf{chall}_2)$
- V sends challenges $(\Delta_j)_{j\in[\tau]} \leftarrow_r [N]^\tau$
- P sends $(open_j, \overline{com}_j)_{j\in[\tau]} \leftarrow \mathsf{VC.Open}(\Delta_j, r_j)$
- V verifies vector commitment, VOLE consistency, QuickSilver proof:
  - $(b_1, (\mathsf{sd}_i^j)_{i\neq\Delta_j, j\in[\tau]}) \leftarrow \mathsf{VC.V}((open_j, \overline{com}_j, \Delta_j)_{j\in[\tau]}, h_{com})$
    ▷ recompute h_{com} and check its consistency, abuse notation of single vector commitment
  - $(b_2, \Delta, \vec{w}) \leftarrow \mathsf{VOLE.V}(\mathsf{chall}_1, t_{\mathsf{check}}, \vec{C}, (\Delta_j, (\mathsf{sd}_i^j)_{i\neq\Delta_j})_{j\in[\tau]}, \ell)$
  - $b_3 \leftarrow \mathsf{V}_{\mathsf{QS}}(\mathcal{C}, \Delta, \vec{w}, \gamma, \pi_{\mathsf{QS}}, \mathsf{chall}_2)$
  - Output $b_1 \wedge b_2 \wedge b_3$

Fig. 4. Zero-knowledge proof for all NP from VOLEitH paradigm

to the standard FS transformation security, we note that in our construction the first challenge is defined by the π_{off} and is generated honestly in advance, therefore, the $\mathcal{A}$ can not make queries to random oracle to cheat for that challenge. We note that the online/offline split does not affect QROM security: the completed proof $(\pi_{\mathsf{off}}, \pi_{\mathsf{on}})$ is identical to the VOLEitH NIZK, so soundness and ZK are inherited from the QROM analyses of [BBB+25] and FAEST v2 (spec §9.7). The only new property, EwR, has a straight-line extractor requiring no rewinding or observation of oracle queries, so it does not weaken QROM security. We in fact gain an advantage: the honest offline phase removes one adversarial Fiat–Shamir round, so only two rounds require QROM analysis.

Theorem 9. *Assume that the vector commitment scheme is computationally hiding, and straight-line extractable binding, hash functions H_{FS} are (programmable) random oracles, QuickSilver proof is SHVZK with negligible soundness error, then the construction in Figure 5 is online/offline NIZK satisfying extraction with randomness. For any p.p.t adversary $\mathcal{A}$ with q_{FS} queries to random oracle H_{FS}, the advantage of $\mathcal{A}$ is bounded by*

$$\mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{A}}(\lambda) \;\le\; q_{\mathsf{FS}}\mathsf{Adv}^{\mathsf{bind}}_{\mathsf{VC}}(\lambda) \;+\; \mathsf{Adv}^{\mathsf{check}}_{\mathsf{VOLE}}(\lambda) \;+\; q_{\mathsf{FS}}\mathsf{Adv}^{\mathsf{snd}}_{\mathsf{QS}}(\lambda),$$

Proof. Completeness. For any $(Y, \vec{y}) \in R$ and randomness r, the offline phase $\mathsf{ONIZK.P}_{\mathsf{off}}(Y; r)$ deterministically define $(r_j)_{j\in[\tau]} \leftarrow \mathsf{PRG}(r)$, computes $(com_j, (\mathsf{sd}_i^j)_{i\in[N]}) \leftarrow \mathsf{VC.Com}(r_j, N)$ for each $j \in [\tau]$, and outputs

$$\pi_{\mathsf{off}} := \big(h_{com}, \vec{C}\big),$$

where $h_{com} := H(com_1, \ldots, com_\tau)$, and $(\vec{u}, \vec{v}, \vec{C}) \leftarrow \mathsf{VOLE.P}\big((\mathsf{sd}_i^j)_{i\in[N],j\in[\tau]}, \ell\big)$.

Given the same r, the online phase $\mathsf{ONIZK.P}_{\mathsf{on}}(Y, \vec{y}, r)$ recomputes the identical VOLE correlation $(\vec{u}, \vec{v}, \vec{C})$, sets $\mathsf{chall}_1 := \mathsf{H}_{\mathsf{FS}}(h_{com}, \vec{C})$, computes $t_{\mathsf{check}} \leftarrow \mathsf{VOLE.C}(\vec{u}, \vec{v}, \mathsf{chall}_1)$ and $\gamma := \vec{y} - \vec{u}_{[0,k-1]}$, defines chall_2 and $(\Delta_j)_{j\in[\tau]}$ via the random oracle, and produces $\pi_{\mathsf{QS}} \leftarrow \mathsf{P}_{\mathsf{QS}}(C, \vec{y}, \vec{u}, \vec{v}, \mathsf{chall}_2)$.
The verifier $\mathsf{ONIZK.V}(Y, \pi)$ recomputes all challenges from π using the same random oracles. Since $(\vec{y}, \vec{u}, \vec{v})$ form a valid VOLE correlation with $\mathcal{C}(\vec{y}) = 1$ and π_{QS} is honestly generated. All three checks, including vector commitment, VOLE consistency, and QuickSilver verification pass with probability 1 by the correctness of VC, VOLE, and QuickSilver.

Adaptive Zero-Knowledge. Since our proof $\pi = (\pi_{\mathsf{on}}, \pi_{\mathsf{off}})$ is constructed from VOLEitH-based interactive proof system and Fiat-Shamir transformation, we refer the reader to the proof security of VOLEitH-based NIZK in [BBD+23, BBB+25] for details. Briefly, adaptive ZK is shown using the hiding property of vector commitment and the SHVZK simulator of QuickSilver where $\mathsf{H}, \mathsf{H}_{\mathsf{FS}}$ are programmable random oracles.

Extraction with Randomness. We assume that the collisions of random oracles for polynomial time of queries is negligible, we now prove that $\mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{A}}(\lambda)$ is negligible for any p.p.t. adversary $\mathcal{A}$. First, we show the correctness of $\mathsf{ONIZK.EwR}$ on honestly generated proofs. By construction (Figure 5), given $\pi = (\pi_{\mathsf{off}}, \pi_{\mathsf{on}})$ and r, the extractor EwR proceeds as follows:

1. Compute $(r_j)_{j\in[\tau]} \leftarrow \mathsf{PRG}(r)$.
2. Recompute $(com_j, (\mathsf{sd}_i^j)_{i\in[N]}) \leftarrow \mathsf{VC.Com}(r_j, N)$ for each j.
3. Recompute $(\vec{u}, \vec{v}, \vec{C}) \leftarrow \mathsf{VOLE.P}((\mathsf{sd}_i^j)_{i\in[N],j\in[\tau]}, \ell)$.
4. Parse γ from π_{on} and output $\vec{y} \leftarrow \gamma + \vec{u}_{[0,k-1]}$.

Since r determines $\vec{u}$ deterministically through PRG, VC.Com, and VOLE.P, and an honest prover sets $\gamma = \vec{y} - \vec{u}_{[0,k-1]}$, the extractor recovers $\vec{y} = \gamma + \vec{u}_{[0,k-1]}$ correctly.

We now show that the advantage of $\mathcal{A}$ is negligible. Suppose for contradiction that $\mathcal{A}$ wins the ewr experiment with non-negligible probability $\epsilon(\lambda)$. That is, $\mathcal{A}$ receives r (which was used to form π_{off}) and outputs π_{on} such that $\mathsf{ONIZK.V}(Y, \pi) = 1$ yet $R(Y, \vec{y}^*) \neq 1$, where $\vec{y}^* = \mathsf{EwR}(\mathsf{crs}, \pi, r)$.

Let $\vec{u}^*$ be the VOLE vector reconstructed from r by EwR, and let γ^* be the value extracted from $\mathcal{A}$'s π_{on}, so that $\vec{y}^* = \gamma^* + \vec{u}^*_{[0,k-1]}$. The winning condition gives

$$\mathsf{ONIZK.V}(Y, \pi) = 1 \quad \text{but} \quad C(\vec{y}^*) \neq 1.$$

Since $\mathsf{ONIZK.V}(Y, \pi) = 1$, the following three checks all pass:

VOLEitH-based ONIZK

Parameters

- $(Y, \vec{y}) \in R$, $\vec{y} \in \{0,1\}^k$, $\mathcal{C}(\vec{y}) = 1$
- System parameters: $\ell \in \mathbb{N}$, $N = 2^n$, $\tau = O(1)$, $N^\tau = 2^\lambda$
- Random oracles $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{2\lambda}$, $\mathsf{H}_{\mathsf{FS}} : \{0,1\}^* \to \{0,1\}^{2\lambda}$

$\pi_{\mathsf{off}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{off}}(Y; r)$

1: $(r_j)_{j\in[\tau]} \leftarrow \mathsf{PRG}(r)$
2: **for** $j \in [\tau]$ **do**
3: $\quad (com_j, (\mathsf{sd}_i^j)_{i\in[N]}) \leftarrow \mathsf{VC.Com}(r_j, N)$
4: $h_{com} := \mathsf{H}(com_1, \cdots, com_\tau)$
5: $(\vec{u}, \vec{v}, \vec{C}) \leftarrow \mathsf{VOLE.P}((\mathsf{sd}_i^j)_{i\in[N],j\in[\tau]}, \ell)$
6: **return** $\pi_{\mathsf{off}} := (h_{com}, \vec{C})$

$\pi_{\mathsf{on}} \leftarrow \mathsf{ONIZK.P}_{\mathsf{on}}(Y, \vec{y}, r)$

1: $(r_j)_{j\in[\tau]} \leftarrow \mathsf{PRG}(r)$
2: **for** $j \in [\tau]$ **do**
3: $\quad (com_j, (\mathsf{sd}_i^j)_{i\in[N]}) \leftarrow \mathsf{VC.Com}(r_j, N)$
4: $h_{com} := \mathsf{H}(com_1, \cdots, com_\tau)$
5: $(\vec{u}, \vec{v}, \vec{C}) \leftarrow \mathsf{VOLE.P}((\mathsf{sd}_i^j)_{i\in[N],j\in[\tau]}, \ell)$
6: $\mathsf{chall}_1 := \mathsf{H}(h_{com}, \vec{C})$
7: $t_{\mathsf{check}} \leftarrow \mathsf{VOLE.C}(\vec{u}, \vec{v}, \mathsf{chall}_1)$
8: $\gamma := \vec{y} - \vec{u}_{[0,k-1]}$
9: $\mathsf{chall}_2 := \mathsf{H}_{\mathsf{FS}}(\mathsf{chall}_1, \gamma, t_{\mathsf{check}})$
10: $\pi_{\mathsf{QS}} \leftarrow \mathsf{P}_{\mathsf{QS}}(Y, \vec{y}, \vec{u}, \vec{v}, \mathsf{chall}_2)$
11: $(\Delta_j)_{j\in[\tau]} \leftarrow \mathsf{H}_{\mathsf{FS}}(\mathsf{chall}_2, \pi_{\mathsf{QS}})$
12: $(open_j, \overline{com}_j)_{j\in[\tau]} \leftarrow \mathsf{VC.Open}(\Delta_j, r_j)$
13: **return** $\pi_{\mathsf{on}} := (t_{\mathsf{check}}, (\Delta_j)_{j\in[\tau]}, \gamma, \pi_{\mathsf{QS}}, (open_j, \overline{com}_j)_{j\in[\tau]})$

$0/1 \leftarrow \mathsf{ONIZK.V}(Y, \pi)$

1: $(h_{com}, \vec{C}, t_{\mathsf{check}}, (\Delta_j)_{j\in[\tau]}, \gamma, \pi_{\mathsf{QS}}, (open_j, \overline{com}_j)_{j\in[\tau]}) \leftarrow \pi$
2: $(b_1, (\mathsf{sd}_i^j)_{i\neq\Delta_j, j\in[\tau]}) \leftarrow \mathsf{VC.V}((open_j, \overline{com}_j, \Delta_j)_{j\in[\tau]}, h_{com})$
3: $\mathsf{chall}_1 := \mathsf{H}_{\mathsf{FS}}(h_{com}, \vec{C})$
4: $(b_2, \Delta, \vec{w}) \leftarrow \mathsf{VOLE.V}(\mathsf{chall}_1, t_{\mathsf{check}}, \vec{C}, (\Delta_j, (\mathsf{sd}_i^j)_{i\neq\Delta_j})_{j\in[\tau]}, \ell)$
5: $\mathsf{chall}_2 := \mathsf{H}_{\mathsf{FS}}(\mathsf{chall}_1, \gamma, t_{\mathsf{check}})$
6: $b_3 \leftarrow \mathsf{V}_{\mathsf{QS}}(Y, \Delta, \vec{w}, \gamma, \pi_{\mathsf{QS}}, \mathsf{chall}_2)$
7: Check $(\Delta_j)_{j\in[\tau]} \stackrel{?}{=} \mathsf{H}_{\mathsf{FS}}(\mathsf{chall}_2, \pi_{\mathsf{QS}})$
8: **return** $b_1 \wedge b_2 \wedge b_3$

$y \leftarrow \mathsf{ONIZK.EwR}(\pi, r)$

1: Parse π to get γ
2: $(r_j)_{j\in[\tau]} \leftarrow \mathsf{PRG}(r)$
3: **for** $j \in [\tau]$ **do**
4: $\quad (com_j, (\mathsf{sd}_i^j)_{i\in[N]}) \leftarrow \mathsf{VC.Com}(r_j, N)$
5: $(\vec{u}, \vec{v}, \vec{C}) \leftarrow \mathsf{VOLE.P}((\mathsf{sd}_i^j)_{i\in[N],j\in[\tau]}, \ell)$
6: **return** $\gamma + \vec{u}_{[0,k-1]}$

Fig. 5. Online/Offline NIZK from VOLEitH

1. *Commitment consistency.* For each $j \in [\tau]$, $\mathsf{VC.V}(\mathsf{open}_j, \Delta_j, com_j)$ accepts and outputs $(\mathsf{sd}_i^j)_{i \neq \Delta_j}$, where $(\Delta_j)_{j\in[\tau]} = \mathsf{H_{FS}}(\mathsf{chall}_2, \pi_{\mathsf{QS}})$.
2. *VOLE consistency.* $\mathsf{VOLE.V}(\mathsf{chall}_1, t_{\mathsf{check}}, \vec{C}, (\Delta_j, (\mathsf{sd}_i^j)_{i\neq\Delta_j})_{j\in[\tau]}, \ell)$ outputs $(\Delta, \vec{w})$ such that the VOLE relation $\vec{w} = \Delta \cdot \vec{u} + \vec{v}$ satisfies.
3. *QuickSilver verification.* $\mathsf{V_{QS}}(Y, \Delta, \vec{w}, \gamma^*, \pi_{\mathsf{QS}}, \mathsf{chall}_2) = 1$ where $\mathsf{chall}_2 := \mathsf{H_{FS}}(\mathsf{chall}_1, \gamma, t_{\mathsf{check}})$.

There are three cases leading to the winning of $\mathcal{A}$.

Case 1: Reduction to extractable binding of VC. In this case, $\mathcal{A}$ produces openings $(\mathsf{open}_j, \Delta_j)_{j\in[\tau]}$ for commitments $(com_j)_{j\in[\tau]}$ that were fixed in π_{off} (determined entirely by r), yet the seeds $(\mathsf{sd}_i^j)_{i\neq\Delta_j}$ recovered by the verifier differ from those generated honestly from r via $\mathsf{VC.Com}(r_j, N)$. Since the challenges $(\Delta_j)_{j\in[\tau]}$ are derived as $\mathsf{H_{FS}}(\mathsf{chall}_2, \pi_{\mathsf{QS}})$ in the random oracle model and are thus unpredictable when the commitments are fixed, $\mathcal{A}$ must have found an alternative opening for some com_j. This contradicts the straight-line extractable binding of VC.

We construct a reduction $\mathcal{B}$ against extractable binding as follow. $\mathcal{B}$ runs $\mathcal{A}$, honestly computes π_{off} from r to obtain $(com_j)_{j\in[\tau]}$, and gives r to $\mathcal{A}$ to obtain π_{on}. If the seeds recovered from $\mathcal{A}$'s openings differ from those derived from r $\mathcal{B}$ outputs, that pair of openings is an output to break binding game. Since $\mathcal{A}$ makes at most q_{FS} queries to the random oracle, by a union bound over all q_{FS} queries, thus

$$\Pr[\text{Case 1}] \;\leq\; q_{\mathsf{FS}}\mathsf{Adv}^{\mathsf{bind}}_{\mathsf{VC}}(\lambda),$$

Case 2: Reduction to failure of consistency check in VOLE construction. In this case (Case 1 ruled out), $\mathcal{A}$'s openings are consistent with the committed seeds. However, from the committed seeds, $\mathcal{A}$ does not follow the VOLE.P to combine and construct the VOLE correlations honestly, i.e., to generate $(\vec{u}, \vec{v}, \vec{C})$, yet it can still pass the VOLE consistency check. Since chall_1 is fixed by π_{off} (determined entirely by r) and is used as the challenge in VOLE.V, $\mathcal{A}$ has no freedom in choosing it. That probability is bounded by the $\mathsf{Adv}^{\mathsf{check}}_{\mathsf{VOLE}}(\lambda)$. We note that in VOLE.V, $(\vec{C}, (\Delta_j, (\mathsf{sd}_i^j)_{i\neq\Delta_j})_{j\in[\tau]})$ is used to define and compute $(\Delta, \vec{w})$, while $(\mathsf{chall}_1, t_{\mathsf{check}})$ is used to check whether $\vec{w} = \Delta \cdot \vec{u} + \vec{v}$.

$$\Pr[\text{Case 2}] \;\leq\; \mathsf{Adv}^{\mathsf{check}}_{\mathsf{VOLE}}(\lambda),$$

Case 3: Reduction to soundness of QuickSilver. In this case (Case 1 and 2 ruled out), $\mathcal{A}$'s openings are consistent with the committed seeds, so the VOLE correlation $(\Delta, \vec{w})$ derived by the verifier satisfies $\vec{w} = \Delta \cdot \vec{u}^* + \vec{v}^*$ with the honest $(\vec{u}^*, \vec{v}^*)$ determined by r. However, $\mathsf{V_{QS}}$ accepts a QuickSilver proof π_{QS} for γ^* using the formed VOLE correlation, while $C(\gamma^* + \vec{u}^*_{[0,k-1]}) \neq 1$. This leads to break the soundness property of QuickSilver proof, i.e., for any fixed global MAC key Δ (determined by the Fiat-Shamir random oracle), the probability that $\mathsf{V_{QS}}$ accepts an incorrect witness is negligible. Since chall_2 is derived from $\mathcal{A}$'s adaptive random oracle queries, by a union bound over all q_{FS} queries, thus

$$\Pr[\text{Case 3}] \;\leq\; q_{\mathsf{FS}}\mathsf{Adv}^{\mathsf{snd}}_{\mathsf{QS}}(\lambda),$$

Therefore, from the three cases, we have:

$$\mathsf{Adv}^{\mathsf{ewr}}_{\mathsf{ONIZK},\mathcal{A}}(\lambda) \;\leq\; q_{\mathsf{FS}}\mathsf{Adv}^{\mathsf{bind}}_{\mathsf{VC}}(\lambda) \;+\; \mathsf{Adv}^{\mathsf{check}}_{\mathsf{VOLE}}(\lambda) \;+\; q_{\mathsf{FS}}\mathsf{Adv}^{\mathsf{snd}}_{\mathsf{QS}}(\lambda),$$

which is negligible. This completes the proof.

6 Implementation and Experimental Results

We implement both the standard and instance-hiding variants of our scheme in Rust. The FAEST v2 Rust implementation[9] is used to instantiate our primitives VOLE, VC, and SIG. The relation $\mathcal{R}$ is instantiated with AES-128 across all variants, this is the standard blockcipher one-wayness relation used in many MPCitH or VOLEitH-based signatures where the extended witness $w_{\mathsf{ext}} = 160B$ and the number of constraints $n_c = 320$. For the instance-hiding variant, we instantiate the hash relation h using RainHash, described in [BBB+26]. RainHash satisfies all the properties required by our instantiation, and the main reason for preferring it over a more standard choice such as SHAKE-256

[9] https://github.com/ait-crypto/faest-rs

is that its circuit size is 10× smaller. In particular, $w_{\mathsf{ext}} = 480B$ and $n_c = 29$ for RainHash; as such, the total is $w_{\mathsf{ext}} = 640$ and $n_c = 349$ for the instance hiding variant since it combines the two. The implementation can be found on: GitHub[10]. We include the dependency manifest (`Cargo.lock`) to ensure reproducible results. All parameters match the FAEST-128f variant which achieves 128-bit security. All experiments were run on a GNU/Linux desktop with an AMD Ryzen 5 5600X processor using a single thread. See Table 2 for key and signature sizes and running times.

To compare our scheme against the state-of-the-art adaptor signatures for NP, we implement the (non-instance-hiding) baseline of Ciampi et al. [CLTZ25] for the same AES-128 relation. As described in Section 2.2, their scheme is built on top of the MPCitH-based verifiable encryption scheme of [TZ24]. For a fair comparison, we instantiate pSig and pVer in Ciampi et al. with the same FAEST-128f scheme, and choose the most compact instantiation of MPCitH-based verifiable encryption from [TZ24], built on the Kyber post-quantum encryption scheme [SAB+20] and the Banquet proof system [BDK+21]. We set the MPCitH-based verifiable encryption parameters to $N = 256$ and $\tau = 16$, which achieves a soundness error of 2^{-128} according to Table 1 of [TZ24]. While our scheme is implemented in Rust, the baseline for Ciampi et al. uses highly optimized C++ implementations of both FAEST-128f[11] and MPCitH verifiable encryption[12]. This implementation asymmetry *favors the baseline*: in particular, the C++ FAEST implementation is over 2× faster than the Rust version, so any running-time advantage we report is conservative. Despite this, our scheme is an order of magnitude faster in Adapt, Ver, and Ext. The slowdown in pSig and pVer can be attributed to the unoptimized FAEST implementation in Rust and the additional cost of running $\mathsf{ONIZK.P}_{\mathsf{off}}$ to generate the offline proof component, which has no counterpart in Ciampi et al. For the concrete implementation of ONIZK using VOLEitH, the pSig of Ciampi et al. is exactly FAEST signing while ours is FAEST signing plus $\mathsf{ONIZK.P}_{\mathsf{off}}$, whose cost is dominated by the vector commitment, accounts for about 50% of FAEST signing time. This overhead is acceptable in practice since pSig is done off-chain; it does not affect any on-chain costs. We expect that our running times can be further improved by switching to the optimized C++ implementation of FAEST-128f and by further optimizing our ONIZK implementation.

Scheme	$\lvert apk\rvert$	$\lvert ask\rvert$	$\lvert\tilde{\sigma}\rvert$	$\lvert\sigma\rvert$	Gen	pSig	pVer	Adapt	Ver	Ext
[CLTZ25] + [TZ24]	32	32	5988	28676	0.00	0.3907	0.3297	55.27	55.55	57.91
Standard (Figure 2)	32	32	5940	11912	0.00	2.248	2.081	1.382	2.401	0.8796
Instance-hiding (Figure 3)	32	32	5972	19592	0.00	2.648	2.497	9.121	9.520	1.073

Table 2. Comparison of key and signature sizes (in bytes) and running times (in milliseconds).

Acknowledgements

Use of AI During the preparation of this work, AI was used to generate the RainHash implementation in Rust, transform the verifiable encryption of [TZ24] into an adaptor signature with FAEST-128f, and generate the benchmark scripts. It also helped us with refactoring, debugging and finding optimization. The authors are solely responsible for the contents and results presented in this paper.

This work was in part supported by JST K Program Grant Number JPMJKP24U2, Japan.

This paper was prepared for information purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall

[10] https://github.com/zama-ai/adaptor-faest/
[11] https://github.com/faest-sign/faest-arch-opt
[12] https://github.com/akiratk0355/verenc-mpcith/tree/main/helium_aes

not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

AEE+21. Lukas Aumayr, Oguzhan Ersoy, Andreas Erwig, Sebastian Faust, Kristina Hostáková, Matteo Maffei, Pedro Moreno-Sanchez, and Siavash Riahi. Generalized channels from limited blockchain scripts and adaptor signatures. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part II*, volume 13091 of *LNCS*, pages 635–664. Springer, Cham, December 2021.

BBB+25. Carsten Baum, Ward Beullens, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Christian Majenz, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. Shorter, tighter, FAESTer: Optimizations and improved (QROM) analysis for VOLE-in-the-head signatures. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 124–156. Springer, Cham, August 2025.

BBB+26. Carsten Baum, Marvin Beckmann, Ward Beullens, Shibam Mukherjee, and Christian Rechberger. Concretely efficient blind signatures based on VOLE-in-the-head proofs and the MAYO trapdoor. Cryptology ePrint Archive, Paper 2026/109, 2026.

BBD+23. Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 581–615. Springer, Cham, August 2023.

BDK+21. Carsten Baum, Cyprien Delpech de Saint Guilhem, Daniel Kales, Emmanuela Orsini, Peter Scholl, and Greg Zaverucha. Banquet: Short and fast signatures from AES. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 266–297. Springer, Cham, May 2021.

Bow16. Sean Bowe. Pay-to-sudoku, 2016. https://github.com/zcash-hackworks/pay-to-sudoku.

CDG+25. Megan Chen, Pousali Dey, Chaya Ganesh, Pratyay Mukherjee, Pratik Sarkar, and Swagata Sasmal. Universally composable non-interactive zero-knowledge from sigma protocols via a new straight-line compiler. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part I*, volume 15674 of *LNCS*, pages 381–417. Springer, Cham, May 2025.

CGGN17. Matteo Campanelli, Rosario Gennaro, Steven Goldfeder, and Luca Nizzardo. Zero-knowledge contingent payments revisited: Attacks and payments for services. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 229–243. ACM Press, October / November 2017.

CLTZ25. Michele Ciampi, Xiangyu Liu, Ioannis Tzannetos, and Vassilis Zikas. Universal adaptor signatures from blackbox multi-party computation. In Arpita Patra, editor, *CT-RSA 2025*, volume 15598 of *LNCS*, pages 375–398. Springer, Cham, April / May 2025.

COSV17. Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Delayed-input non-malleable zero knowledge and multi-party coin tossing in four rounds. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 711–742. Springer, Cham, November 2017.

CPS+16a. Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Improved OR-composition of sigma-protocols. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 112–141. Springer, Berlin, Heidelberg, January 2016.

CPS+16b. Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Online/offline OR composition of sigma protocols. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 63–92. Springer, Berlin, Heidelberg, May 2016.

CPV20. Michele Ciampi, Roberto Parisella, and Daniele Venturi. On adaptive security of delayed-input sigma protocols and Fiat-Shamir NIZKs. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 20*, volume 12238 of *LNCS*, pages 670–690. Springer, Cham, September 2020.

DFMS22. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Efficient NIZKs and signatures from commit-and-open protocols in the QROM. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 729–757. Springer, Cham, August 2022.

DOY22. Wei Dai, Tatsuaki Okamoto, and Go Yamamoto. Stronger security and generic constructions for adaptor signatures. In Takanori Isobe and Santanu Sarkar, editors, *INDOCRYPT 2022*, volume 13774 of *LNCS*, pages 52–77. Springer, Cham, December 2022.

EEE20. Muhammed F. Esgin, Oguzhan Ersoy, and Zekeriya Erkin. Post-quantum adaptor signatures and payment channel networks. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *ESORICS 2020, Part II*, volume 12309 of *LNCS*, pages 378–397. Springer, Cham, September 2020.

EFH+21. Andreas Erwig, Sebastian Faust, Kristina Hostáková, Monosij Maitra, and Siavash Riahi. Two-party adaptor signatures from identification schemes. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 451–480. Springer, Cham, May 2021.

Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, Berlin, Heidelberg, August 2005.

Fuc19. Georg Fuchsbauer. WI is not enough: Zero-knowledge contingent (service) payments revisited. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 49–62. ACM Press, November 2019.

GMM+22. Noemi Glaeser, Matteo Maffei, Giulio Malavolta, Pedro Moreno-Sanchez, Erkan Tairi, and Sri Aravinda Krishnan Thyagarajan. Foundations of coin mixing services. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 1259–1273. ACM Press, November 2022.

GOS06. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Berlin, Heidelberg, May / June 2006.

Gro06. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Berlin, Heidelberg, December 2006.

GRS25. Paul Gerhart, Daniel Rausch, and Dominique Schröder. Universally composable adaptor signatures. Cryptology ePrint Archive, Report 2025/1363, 2025.

GSST24. Paul Gerhart, Dominique Schröder, Pratik Soni, and Sri Aravinda Krishnan Thyagarajan. Foundations of adaptor signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 161–189. Springer, Cham, May 2024.

Kat21. Shuichi Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 580–610, Virtual Event, August 2021. Springer, Cham.

KKW18. Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 525–537. ACM Press, October 2018.

LTZ24. Xiangyu Liu, Ioannis Tzannetos, and Vassilis Zikas. Adaptor signatures: New security definition and a generic construction for NP relations. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part II*, volume 15485 of *LNCS*, pages 168–193. Springer, Singapore, December 2024.

MB26. Ayush Meshram and Ayush Banerjee. Multivariate witness-hiding adaptor signatures. Cryptology ePrint Archive, Paper 2026/690, 2026.

MBB+19. Andrew Miller, Iddo Bentov, Surya Bakshi, Ranjit Kumaresan, and Patrick McCorry. Sprites and state channels: Payment networks that go faster than lightning. In Ian Goldberg and Tyler Moore, editors, *FC 2019*, volume 11598 of *LNCS*, pages 508–526. Springer, Cham, February 2019.

MMK+17. Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Srivatsan Ravi. Concurrency and privacy with payment-channel networks. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 455–471. ACM Press, October / November 2017.

MMS+19. Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. Anonymous multi-hop locks for blockchain scalability and interoperability. In *NDSS 2019*. The Internet Society, February 2019.

MTV+23. Varun Madathil, Sri Aravinda Krishnan Thyagarajan, Dimitrios Vasilopoulos, Lloyd Fournier, Giulio Malavolta, and Pedro Moreno-Sanchez. Cryptographic oracle-based conditional payments. In *NDSS 2023*. The Internet Society, February 2023.

NAA20. Ky Nguyen, Miguel Ambrona, and Masayuki Abe. WI is almost enough: Contingent payment all over again. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 641–656. ACM Press, November 2020.

Poe17. Andrew Poelstra. Scriptless scripts, 2017. https://download.wpsoftware.net/bitcoin/wizardry/mw-slides/2017-03-mit-bitcoin-expo/slides.pdf.

QPM+23. Xianrui Qin, Shimin Pan, Arash Mirzaei, Zhimei Sui, Oguzhan Ersoy, Amin Sakzad, Muhammed F. Esgin, Joseph K. Liu, Jiangshan Yu, and Tsz Hon Yuen. BlindHub: Bitcoin-compatible privacy-preserving payment channel hubs supporting variable amounts. In *2023 IEEE Symposium on Security and Privacy*, pages 2462–2480. IEEE Computer Society Press, May 2023.

Roy22. Lawrence Roy. SoftSpokenOT: Quieter OT extension from small-field silent VOLE in the minicrypt model. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 657–687. Springer, Cham, August 2022.

RT25. Lior Rotem and Stefano Tessaro. Straight-line knowledge extraction for multi-round protocols. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 95–127. Springer, Cham, August 2025.

SAB+20. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

Sch93. Claus-Peter Schnorr. FFT-hash II, efficient cryptographic hashing. In Rainer A. Rueppel, editor, *EUROCRYPT'92*, volume 658 of *LNCS*, pages 45–54. Springer, Berlin, Heidelberg, May 1993.

TMSM21. Erkan Tairi, Pedro Moreno-Sanchez, and Matteo Maffei. Post-quantum adaptor signature for privacy-preserving off-chain payments. In *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part II*, page 131–150, Berlin, Heidelberg, 2021. Springer-Verlag.

TZ24. Akira Takahashi and Greg Zaverucha. Verifiable encryption from MPC-in-the-head. *CiC*, 1(1):3, 2024.

YSWW21. Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. QuickSilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2986–3001. ACM Press, November 2021.

A Related Notions

A.1 Security Notions for NIZKs

Definition 12 (Completeness). *A non-interactive proof system for relation R is complete if, for all $\lambda \in \mathbb{N}$, $(Y, y) \in \mathcal{LW}_R(\lambda)$, for $crs \leftarrow \mathsf{Setup}(1^\lambda)$ and $\pi \in \mathsf{P}(crs, Y, y)$, it holds that $\mathsf{V}(crs, Y, \pi) = 1$.*

Definition 13 (Soundness). *A non-interactive proof system for relation R is sound if, for all p.p.t. algorithm $\mathcal{A}$, probability*

$$\mathsf{Adv}^{\mathsf{snd}}_{NIZK,\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l|l} Y \notin \mathcal{L}_R(\lambda) \wedge & crs \leftarrow \mathsf{Setup}(1^\lambda) \\ 1 = \mathsf{V}(crs, Y, \pi) & (Y, \pi) \leftarrow \mathcal{A}(crs) \end{array}\right]$$

is negligible.

A.2 Definitions for Digital Signature Schemes

Definition 14 (Digital Signature Scheme). *A digital signature scheme SIG is a tuple of algorithms $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ such that:*

- $\mathsf{Gen}(1^\lambda)$ *outputs a key pair* $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{Sig}(\mathsf{sk}, m)$ *outputs a signature σ on message m using secret key* sk.
- $\mathsf{Ver}(\mathsf{pk}, m, \sigma)$ *outputs 1 if σ is a valid signature on m under public key* pk, *and 0 otherwise.*

It must satisfy the correctness condition: for all $\lambda \in \mathbb{N}$, $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{Gen}(1^\lambda)$, $m \in \{0,1\}^*$, and $\sigma \in \mathsf{Sig}(\mathsf{sk}, m)$, it holds that $\mathsf{Ver}(\mathsf{pk}, m, \sigma) = 1$.

Definition 15 (Existential Unforgeability under Chosen Message Attack: EUF-CMA). *A digital signature scheme SIG is existentially unforgeable under chosen message attack (EUF-CMA secure) if for any p.p.t. adversary $\mathcal{A}$, advantage $\mathsf{Adv}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) := \Pr[\mathrm{Expr}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) \to 1]$ is negligible in λ, where $\mathrm{Expr}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$ is defined as follows.*

$\underline{\mathrm{Expr}^{\mathsf{cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)}$
1: $Q_{\mathsf{Sig}} := \emptyset$
2: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda)$
3: $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OSig}(\cdot)}(\mathsf{pk})$
4: $b := (\mathsf{Ver}(\mathsf{pk}, m^*, \sigma^*) = 1 \wedge m^* \notin Q_{\mathsf{Sig}})$
5: **return** b

$\underline{\mathsf{OSig}(m)}$
1: $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, m)$
2: $Q_{\mathsf{Sig}} := Q_{\mathsf{Sig}} \cup \{m\}$
3: Return σ

A.3 Other Security Notions for Adaptor Signatures

Unforgeability. Adaptor signatures are unforgeable if no adversary can create an adapted signature on a fresh message never asked for (pre-)signing or adapt a pre-signature on an instance whose witness is not known. Due to the second requirement, this notion naturally demans the considered relation be hard.

Definition 16 (Unforgeability [LTZ24]). *An adaptor signature scheme AS for hard relation R is unforgeable under chosen message attack (UF-CMA secure), if, for any p.p.t. adversary $\mathcal{A}$, advantage $\mathsf{Adv}^{\mathsf{unf}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \Pr[\mathrm{Expr}^{\mathsf{unf}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1]$ is negligible over λ, where $\mathrm{Expr}^{\mathsf{unf}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ is:*

$\underline{\mathrm{Expr}^{\mathsf{unf}}_{\mathsf{AS},\mathcal{A}}(\lambda)}$
1: $Q_{\mathsf{Sig}} := \emptyset,\ Q_{\mathsf{pSig}} := \emptyset,\ Q_{\mathsf{Sample}} := \emptyset$
2: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$
3: $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OSig}(\cdot),\mathsf{OpSig}(\cdot),\mathsf{OSample}(1^\lambda)}(apk)$
4: $b_1 := (\mathsf{Ver}(apk, m^*, \sigma^*) = 1 \wedge m^* \notin Q_{\mathsf{Sig}})$
5: $b_{21} := (Q_{\mathsf{pSig}}[m^*] = \emptyset)$
6: $b_{22} := (\forall\, (Y, \tilde{\sigma}) \in Q_{\mathsf{pSig}}[m^*],\ Y \in Q_{\mathsf{Sample}})$
7: **return** $b_1 \wedge (b_{21} \vee b_{22})$

$\underline{\mathsf{OSig}(m)}$
1: $Q_{\mathsf{Sig}} := Q_{\mathsf{Sig}} \cup \{m\}$
2: **return** $\mathsf{Sig}(ask, m)$

$\underline{\mathsf{OpSig}(m, Y)}$
1: $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$
2: $Q_{\mathsf{pSig}}[m] := Q_{\mathsf{pSig}}[m] \cup \{(Y, \tilde{\sigma})\}$
3: **return** $\tilde{\sigma}$

$\underline{\mathsf{OSample}(1^\lambda)}$
1: $Y \leftarrow \mathsf{Sample}_{\mathcal{L}_R}(1^\lambda)$
2: $Q_{\mathsf{Sample}} := Q_{\mathsf{Sample}} \cup \{Y\}$
3: **return** Y

Definition 17 ((Perfect) Pre-signature Adaptability [LTZ24, GSST24]). *An adaptor signature scheme* AS *for relation* R *is pre-signature adaptable, if for any* $(Y, y) \in \mathcal{LW}_R(\lambda)$ *and any* apk, m, *and* $\tilde{\sigma}$ *satisfying* $\mathsf{pVer}(apk, m, Y, \tilde{\sigma}) = 1$, *it holds that* $\mathsf{Ver}(apk, m, \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)) = 1$.

The above definition taken from [LTZ24, GSST24] considers malicious keys. It therefore expects that pVer checks the correctness of given apk, which is often implicit.

Definition 18 (AS Witness Hiding [LTZ24]). *An adaptor signature scheme* AS *for relation* R *is witness hiding, if there exists a p.p.t. computable function* $\mathsf{f_{ck}}$ *and a p.p.t. simulator* Sim *such that, for any p.p.t. adversary* $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{wh}}_{\mathsf{AS},\mathsf{Sim},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{Expr}^{\mathsf{wh},0}_{\mathsf{AS},\mathsf{Sim},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{wh},1}_{\mathsf{AS},\mathsf{Sim},\mathcal{A}}(\lambda) \to 1]\right|$$

is negligible in λ, *where* $\mathrm{Expr}^{\mathsf{wh},\mathsf{b}}_{\mathsf{AS},\mathsf{Sim},\mathcal{A}}(\lambda)$ *for* $b \in \{0,1\}$ *are defined as follows.*

$\underline{\mathrm{Expr}^{\mathsf{wh},\mathsf{b}}_{\mathsf{AS},\mathsf{Sim},\mathcal{A}}(\lambda)}$
1: $(Y, y) \leftarrow \mathsf{Sample}(1^\lambda)$
2: $\tilde{b} \leftarrow \mathcal{A}^{\mathsf{O_b}(\cdot,\cdot,\cdot)}(Y)$
3: ***return*** $\tilde{b}$

$\underline{\mathsf{O}_0(apk, ask, m)}$
1: $\mathsf{Assert}(\mathsf{f_{ck}}(apk, ask) = 1)$
2: $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$
3: $\sigma \leftarrow \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)$
4: ***return*** σ

$\underline{\mathsf{O}_1(apk, ask, m)}$
1: $\mathsf{Assert}(\mathsf{f_{ck}}(apk, ask) = 1)$
2: $\sigma \leftarrow \mathsf{Sim}(ask, m, Y)$
3: ***return*** σ

Remark 1. The adversary does not have access to the witness. Consequently, an entity with knowledge of the witness may be able to distinguish adapted signatures from simulated ones. This could be an issue when relation R has disjunctive structure, meaning an instance admits multiple distinct witnesses (e.g., y_0 and y_1). An entity knowing y_0 could distinguish a signature adapted using y_1 from the simulation.

Remark 2. In [LTZ24], function $\mathsf{f_{ck}}$ is not formally defined, but it must not be trivial, e.g., returning 0 for all inputs, to make the above definition meaningful.

Pre-verify soundness, introduced in [GSST24], requires that no adversary, even with the legitimate signing key, can forge a pre-signature on an instance that does not belong to the language. This notion complements pre-signature adaptability (Definition 4) where only valid instances are considered. It is required for some applications such as coin-mixing protocols [GSST24] where the instance is chosen by the signer and the verifier needs to ensure the instance is valid before accepting the pre-signature.

Pre-verify soundness comes in two flavors: computational and statistical, depending on the strength of the adversary. We present both definitions below.

Definition 19 (Computational Pre-Verify Soundness [GSST24]). *An adaptor signature scheme* AS *for relation* R *satisfies computational pre-verify soundness if for every p.p.t. adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that for every* $\lambda \in \mathbb{N}$ *and* $Y \in \{0,1\}^\lambda \setminus \mathcal{L}_R(\lambda)$,

$$\Pr\left[(ask, apk) \leftarrow \mathsf{Gen}(1^\lambda),\ (m, \tilde{\sigma}) \leftarrow \mathcal{A}(ask) : \mathsf{pVer}(apk, m, \tilde{\sigma}, Y) = 1\right] \leq \nu(\lambda)$$

Definition 20 (Statistical Pre-Verify Soundness [GSST24]). *An adaptor signature scheme* AS *for relation* R *satisfies statistical pre-verify soundness if for every* $\lambda \in \mathbb{N}$, $Y \in \{0,1\}^\lambda \setminus \mathcal{L}_R(\lambda)$, *and key pair* $(apk, ask) \in \mathsf{Gen}(1^\lambda)$,

$$\Pr\left[\exists\ (m, \tilde{\sigma})\ :\ \mathsf{pVer}(apk, m, \tilde{\sigma}, Y) = 1\right] = 0$$

As noted in [GSST24], pre-verify soundness is similar to the soundness of a non-interactive proof system, where the pre-signature serves as the proof and the pre-verification algorithm acts as the verifier. When the language induced from the relation is trivial, pre-verification soundness is indeed achievable simply by letting the pre-verification algorithm check the validity of the instance. More generally, computational pre-verify soundness can be achieved by requiring the pre-verification algorithm to check the validity of the instance using a transparent NIZK. Since our construction bases on NIZK, this feature is naturally integrated in our schemes in exchange of the cost for proving additional statement.

The universal composability (UC) of adaptor signatures is studied in [GRS25], which also introduces additional security notions such as strong pre-signature unforgeability that are required to achieve UC security. These stronger guarantees are beyond the scope of this work; in particular, strong pre-signature unforgeability is not required by any of the standard security notions we consider. Nevertheless, our construction can achieve it under the assumption that it is hard to find a distinct randomness that produces a given offline proof.

A.4 Variations of Instance Hiding Notions

Since the witness hiding is defined in the simulation paradigm in Definition 5, it is natural to think about instance hiding notion in the simulation paradigm as well. The following formal definition differs from that of aWH only in that SimSig is not given instance Y.

Definition 21 (Simulation-based Instance Hiding). *An adaptor signature scheme* AS *for relation* R *is simulation-based instance hiding (*IH-Sim*), if there exists a pair of p.p.t. algorithms* $\mathsf{Sim} := (\mathsf{SimGen}, \mathsf{SimSig})$ *that, for any p.p.t. stateful adversary* $\mathcal{A}$*, advantage*

$$\mathsf{Adv}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda) \to 1]\right|$$

is negligible in λ*, where* $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ *and* $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)$ *are defined as follows.*

$\underline{\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{A}}(\lambda)}$
1: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(Y, y, m, \mathsf{state}) \leftarrow \mathcal{A}(ask)$
3: $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y)$
4: $\sigma \leftarrow \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)$
5: $b_1 := \mathcal{A}(\mathsf{state}, \sigma)$
6: $b_2 := (R(Y, y) = 1)$
7: **return** $b_1 \wedge b_2$

$\underline{\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)}$
1: $(apk, ask, \tau) \leftarrow \mathsf{SimGen}(1^\lambda)$
2: $(Y, y, m, \mathsf{state}) \leftarrow \mathcal{A}(ask)$
3: $\sigma \leftarrow \mathsf{SimSig}(ask, m, \tau)$
4: $b_1 := \mathcal{A}(\mathsf{state}, \sigma)$
5: $b_2 := (R(Y, y) = 1)$
6: **return** $b_1 \wedge b_2$

A further natural variant, analogous to the left-or-right (LOR) formulation of indistinguishability for encryption, lets the adversary pick two instance/witness pairs up front and challenges it to tell which of the two is actually used to produce the adapted signature it sees.

Definition 22 (Left-or-Right Instance Hiding). *An adaptor signature scheme* AS *for relation* R *is left-or-right instance hiding (*IH-LOR*), if, for any p.p.t. stateful adversary* $\mathcal{A}$*, advantage*

$$\mathsf{Adv}^{\mathsf{ih\text{-}lor}}_{\mathsf{AS},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}lor},0}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] - \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}lor},1}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1]\right|$$

is negligible in λ*, where* $\mathrm{Expr}^{\mathsf{ih\text{-}lor},\beta}_{\mathsf{AS},\mathcal{A}}(\lambda)$ *for* $\beta \in \{0, 1\}$ *is defined as follows.*

$\underline{\mathrm{Expr}^{\mathsf{ih\text{-}lor},\beta}_{\mathsf{AS},\mathcal{A}}(\lambda)}$
1: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(Y_0, y_0, Y_1, y_1, m, \mathsf{state}) \leftarrow \mathcal{A}(ask)$
3: $\tilde{\sigma} \leftarrow \mathsf{pSig}(ask, m, Y_\beta)$
4: $\sigma \leftarrow \mathsf{Adapt}(apk, m, \tilde{\sigma}, y_\beta)$
5: $b_1 := \mathcal{A}(\mathsf{state}, \sigma)$
6: $b_2 := (R(Y_0, y_0) = 1 \wedge R(Y_1, y_1) = 1)$

7: **return** $b_1 \wedge b_2$

We now show that these notions form a chain $\mathsf{IH} \Rightarrow \mathsf{IH\text{-}Sim} \Rightarrow \mathsf{IH\text{-}LOR}$.

Theorem 10 ($\mathsf{IH} \Rightarrow \mathsf{IH\text{-}Sim}$). *If an adaptor signature scheme* AS *is instance hiding, then it is simulation-based instance hiding.*

Proof. We construct a simulator, $\mathsf{Sim} := (\mathsf{SimGen}, \mathsf{SimSig})$ as follows. Define $\mathsf{SimGen}(1^\lambda) := (apk, ask, \bot)$ where $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$ (the trapdoor is unused), and $\mathsf{SimSig}(ask, m, \tau) := \mathsf{Sig}(ask, m)$ (which ignores τ).

With this construction, $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ (Definition 21) runs $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$, obtains $(Y, y, m, \mathsf{state}) \leftarrow \mathcal{A}(ask)$, sets $\sigma \leftarrow \mathsf{Adapt}(apk, m, \mathsf{pSig}(ask, m, Y), y)$, and returns $\mathcal{A}(\mathsf{state}, \sigma) \wedge (R(Y, y) = 1)$. This results in the steps of $\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ defined to coincide with $\mathrm{Expr}^{\mathsf{awh\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ (Definition 5). Similarly, $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)$ runs $(apk, ask) \leftarrow \mathsf{SimGen}(1^\lambda) = \mathsf{Gen}(1^\lambda)$ and sets $\sigma \leftarrow \mathsf{SimSig}(ask, m, \tau) = \mathsf{Sig}(ask, m)$, which are exactly the steps of $\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda)$, obtained from $\mathrm{Expr}^{\mathsf{awh\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda)$ by replacing $(\mathsf{SimGen}, \mathsf{SimSig})$ with $(\mathsf{Gen}, \mathsf{Sig})$ (Definition 5). Hence

$$\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] = \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}adapt}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1] \text{ and } \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{A}}(\lambda) \to 1] = \Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sig}}_{\mathsf{AS},\mathcal{A}}(\lambda) \to 1],$$

so that $\mathsf{Adv}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{ih}}_{\mathsf{AS},\mathcal{A}}(\lambda)$. Since the right-hand side is negligible by assumption, so is the left-hand side, showing AS is $\mathsf{IH\text{-}Sim}$ secure. □

Theorem 11 ($\mathsf{IH\text{-}Sim} \Rightarrow \mathsf{IH\text{-}LOR}$). *If an adaptor signature scheme* AS *is simulation-based instance hiding, then it is left-or-right instance hiding.*

Proof. Let $\mathsf{Sim} := (\mathsf{SimGen}, \mathsf{SimSig})$ be the pair of p.p.t. algorithms witnessing that AS is $\mathsf{IH\text{-}Sim}$ secure (Definition 21), and let $\mathcal{A}$ be any p.p.t. stateful $\mathsf{IH\text{-}LOR}$ adversary. We relate $\mathrm{Expr}^{\mathsf{ih\text{-}lor},0}_{\mathsf{AS},\mathcal{A}}(\lambda)$ and $\mathrm{Expr}^{\mathsf{ih\text{-}lor},1}_{\mathsf{AS},\mathcal{A}}(\lambda)$ (Definition 22) through the following sequence of games.

- Game 0: This is $\mathrm{Expr}^{\mathsf{ih\text{-}lor},0}_{\mathsf{AS},\mathcal{A}}(\lambda)$: $(apk, ask) \leftarrow \mathsf{Gen}(1^\lambda)$; $(Y_0, y_0, Y_1, y_1, m, \mathsf{state}) \leftarrow \mathcal{A}(ask)$; $\sigma \leftarrow \mathsf{Adapt}(apk, m, \mathsf{pSig}(ask, m, Y_0), y_0)$; return $\mathcal{A}(\mathsf{state}, \sigma) \wedge (R(Y_0, y_0) = 1) \wedge (R(Y_1, y_1) = 1)$.
- Game 1: As Game 0, except $(apk, ask, \tau) \leftarrow \mathsf{SimGen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{SimSig}(ask, m, \tau)$ replace, respectively, the key generation and the pre-sign-and-adapt step.
- Game 2: This is $\mathrm{Expr}^{\mathsf{ih\text{-}lor},1}_{\mathsf{AS},\mathcal{A}}(\lambda)$, i.e., Game 0 with $\sigma \leftarrow \mathsf{Adapt}(apk, m, \mathsf{pSig}(ask, m, Y_1), y_1)$ in place of the challenge signature.

Let P_i denote the probability that Game i returns 1. By definition, $\mathsf{Adv}^{\mathsf{ih\text{-}lor}}_{\mathsf{AS},\mathcal{A}}(\lambda) = |P_0 - P_2|$.

We bound $|P_0 - P_1|$ by constructing an $\mathsf{IH\text{-}Sim}$ adversary $\mathcal{B}_0$ as follows. Given ask, $\mathcal{B}_0$ runs $\mathcal{A}(ask)$ to obtain $(Y_0, y_0, Y_1, y_1, m, \mathsf{state})$, computes $c := (R(Y_1, y_1) = 1)$ (efficiently checkable since R is efficiently verifiable given a witness), and outputs $(Y_0, y_0, m, (\mathsf{state}, c))$ to its own challenger. On receiving $((\mathsf{state}, c), \sigma)$ back, $\mathcal{B}_0$ runs $\mathcal{A}(\mathsf{state}, \sigma)$ to obtain b'_1 and outputs $b'_1 \wedge c$. If $\mathcal{B}_0$ is playing $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{B}_0}(\lambda)$, then σ is generated exactly as in Game 0 and $\mathcal{B}_0$'s output equals $b'_1 \wedge c \wedge (R(Y_0, y_0) = 1)$, which is Game 0's return value; if $\mathcal{B}_0$ is playing $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{B}_0}(\lambda)$, then σ is generated exactly as in Game 1, and likewise $\mathcal{B}_0$'s output equals Game 1's return value. Hence $\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{B}_0}(\lambda) \to 1] = P_0$, $\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{B}_0}(\lambda) \to 1] = P_1$, and $|P_0 - P_1| = \mathsf{Adv}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{B}_0}(\lambda)$.

Symmetrically, define $\mathcal{B}_1$ by swapping the roles of the two pairs: given ask, it runs $\mathcal{A}(ask)$, computes $c := (R(Y_0, y_0) = 1)$, outputs $(Y_1, y_1, m, (\mathsf{state}, c))$, and on $((\mathsf{state}, c), \sigma)$ outputs $\mathcal{A}(\mathsf{state}, \sigma) \wedge c$. As above, $\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{B}_1}(\lambda) \to 1] = P_2$. Moreover, since $\mathsf{SimSig}(ask, m, \tau)$ does not take Y as input, $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{B}_1}(\lambda)$ runs exactly the same steps as $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{B}_0}(\lambda)$: both sample $(apk, ask, \tau) \leftarrow \mathsf{SimGen}(1^\lambda)$, invoke $\mathcal{A}(ask)$ once to obtain the same $(Y_0, y_0, Y_1, y_1, m, \mathsf{state})$, draw $\sigma \leftarrow \mathsf{SimSig}(ask, m, \tau)$, and return $\mathcal{A}(\mathsf{state}, \sigma) \wedge (R(Y_0, y_0) = 1) \wedge (R(Y_1, y_1) = 1)$, i.e., Game 1's return value. Hence $\Pr[\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim},\mathcal{B}_1}(\lambda) \to 1] = P_1$ as well, giving $|P_1 - P_2| = \mathsf{Adv}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{B}_1}(\lambda)$.

In summary, it holds that

$$\mathsf{Adv}^{\mathsf{ih\text{-}lor}}_{\mathsf{AS},\mathcal{A}}(\lambda) = |P_0 - P_2| \le |P_0 - P_1| + |P_1 - P_2| = \mathsf{Adv}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{B}_0}(\lambda) + \mathsf{Adv}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS},\mathcal{B}_1}(\lambda).$$

Both $\mathcal{B}_0$ and $\mathcal{B}_1$ run in time polynomial in that of $\mathcal{A}$, so, since AS is $\mathsf{IH\text{-}Sim}$ secure, both terms on the right are negligible in λ. Hence $\mathsf{Adv}^{\mathsf{ih\text{-}lor}}_{\mathsf{AS},\mathcal{A}}(\lambda)$ is negligible for every p.p.t. $\mathcal{A}$. □

Immediately from Theorems 10 and 11, we have:

Corollary 1 (IH $\Rightarrow$ IH-LOR). *If an adaptor signature scheme is instance hiding, then it is left-or-right instance hiding.*

Remark 3. The converse of the corollary above does not hold: IH-LOR only requires σ to hide which of the two adversarially-chosen instances was used to produce it, whereas IH (and IH-Sim) requires σ to be indistinguishable from a signature produced without any instance at all. This mirrors an observation for the encryption analogues of these notions: unlike left-or-right and real-or-random indistinguishability for encryption, which both compare outputs of the same encryption algorithm, IH and IH-Sim compare outputs of two different algorithms, namely $\mathsf{Adapt}(\cdot,\cdot,\mathsf{pSig}(\cdot,\cdot,\cdot),\cdot)$ and Sig (resp. SimSig).

The gap noted in Remark 3 is not merely a proof artifact: neither IH-Sim nor IH-LOR implies unlinkability (Definition 6), even though IH does (Theorem 1). Intuitively, both IH-Sim and IH-LOR only ever compare an adapted signature ($\mathsf{Adapt}^\star$'s output, tagged 1) against another output that is tagged the *same* way. They never compare against a genuinely-signed $\mathsf{Sig}^\star(ask, m)$ (tagged 0), which is the one comparison unlinkability actually demands. Our formal claim follows.

Proposition 1 (IH-Sim(IH-LOR) $\not\Rightarrow$ UL). *Let* $\mathsf{AS} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{pSig}, \mathsf{Adapt}, \mathsf{Ver})$ *be any simulation-based instance hiding adaptor signature scheme for relation* R*. Then there is an adaptor signature scheme* $\mathsf{AS}^\star$ *for* R *that is simulation-based instance hiding (and left-or-right instance hiding) but not unlinkable.*

Proof. Since AS is IH-Sim secure, there exists a simulator, $\mathsf{Sim} := (\mathsf{SimGen}, \mathsf{SimSig})$. Define $\mathsf{AS}^\star := (\mathsf{Gen}, \mathsf{Sig}^\star, \mathsf{pSig}, \mathsf{Adapt}^\star, \mathsf{Ver}^\star)$ by tagging every signature with the algorithm that produced it:

$$\mathsf{Sig}^\star(ask, m) := (0, \mathsf{Sig}(ask, m)), \qquad \mathsf{Adapt}^\star(apk, m, \tilde{\sigma}, y) := (1, \mathsf{Adapt}(apk, m, \tilde{\sigma}, y)),$$

and $\mathsf{Ver}^\star(apk, m, (t, \sigma)) := \mathsf{Ver}(apk, m, \sigma)$ for $t \in \{0, 1\}$. Namely, the verification ignores the tag. Since the tag is a public, deterministic bit that neither algorithm needs the secret key or a witness to compute, and $\mathsf{Ver}^\star$ discards it before checking σ against Ver, $\mathsf{AS}^\star$ inherits correctness, unforgeability, and pre-signature adaptability from AS.

To see that $\mathsf{AS}^\star$ is not unlinkable, observe that, on any query (m, Y, y) to O_b (Definition 6), the returned signature is tagged 1 if $b = 0$ and tagged 0 if $b = 1$. So, the adversary outputs the tag it receives and it results in $\mathsf{Adv}^{\mathsf{ul}}_{\mathsf{AS}^\star,\mathcal{A}}(\lambda) = 1$.

Nonetheless, $\mathsf{AS}^\star$ is IH-Sim secure with simulator $\mathsf{Sim}^\star := (\mathsf{SimGen}, \mathsf{SimSig}^\star)$, where $\mathsf{SimSig}^\star(ask, m, \tau) := (1, \mathsf{SimSig}(ask, m, \tau))$. Observe that $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{AS}^\star,\mathcal{A}}(\lambda)$ returns $(1, \mathsf{Adapt}(apk, m, \mathsf{pSig}(ask, m, Y), y))$ and $\mathrm{Expr}^{\mathsf{ih\text{-}sim}}_{\mathsf{Sim}^\star,\mathcal{A}}(\lambda)$ returns $(1, \mathsf{SimSig}(ask, m, \tau))$. Namely, the tag is the constant 1 in both games. Accordingly, any adversary's advantage against $\mathsf{AS}^\star$ equals its advantage against AS (Definition 21), which is negligible by assumption. By Theorem 11, $\mathsf{AS}^\star$ is then also IH-LOR secure. □