Adaptor Signatures: New Security Definition and A Generic Construction for NP Relations*

Xiangyu Liu[1,2], Ioannis Tzannetos[1,3], and Vassilis Zikas[2]

[1] Purdue University
[2] Georgia Institute of Technology
[3] National Technical University of Athens
liu3894@purdue.edu, itzannet@purdue.edu, vzikas@gatech.edu

Abstract. An adaptor signatures (AS) scheme is an extension of digital signatures that allows the signer to generate a *pre-signature* for an instance of a hard relation. This pre-signature can later be adapted to a full signature with a corresponding witness. Meanwhile, the signer can extract a witness from both the pre-signature and the signature. AS have recently garnered more attention due to its scalability and interoperability. Dai *et al.* [INDOCRYPT 2022] proved that AS can be constructed for any NP relation using a generic construction. However, their construction has a shortcoming: the associated witness is exposed by the adapted signature. This flaw poses limits the applications of AS, even in its motivating setting, i.e., blockchain, where the adapted signature is typically uploaded to the blockchain and is public to everyone.

To address this issue, in this work we augment the security definition of AS by a natural property which we call *witness hiding*. We then prove the existence of AS for any NP relation, assuming the existence of one-way functions. Concretely, we propose a generic construction of witness-hiding AS from signatures and a weak variant of trapdoor commitments, which we term *trapdoor commitments with a specific adaptable message*. We instantiate the latter based on the Hamiltonian cycle problem. Since the Hamiltonian cycle problem is NP-complete, we can obtain witness hiding adaptor signatures for any NP relation.

1 Introduction

Blockchain technology has emerged as a disruptor, offering decentralized frameworks for various applications. Each transaction on the blockchain operates within a scripting language validated by nodes through a decentralized consensus protocol. Cryptocurrencies like Bitcoin and Ethereum utilize blockchain technologies to power their operations. However, executing transactions on blockchains often incurs significant costs, as users are required to pay fees to entities that run the consensus protocol. These

* Work done while the authors were at Purdue University.

fees are determined by the storage and computational costs associated with transaction scripts. To mitigate this issue, the utilization of adaptor signatures has been proposed as a means to reduce on-chain fees paid to nodes in a wide range of decentralized finance (DeFi) applications (see some examples below).

The notation of adaptor signatures (AS, a.k.a. scriptless scripts) was proposed by Poelstra in 2017 [21, 22] and later formalized by Aumayr *et al.* [2, 3]. An AS scheme is related to a hard relation R such that the signer, holding the signing secret key, can pre-sign a message (e.g., a transaction) with respect to some instance Y to obtain a pre-signature $\tilde{\sigma}$, which can later be adapted to a full signature σ with the knowledge of y, the witness of Y such that $(Y, y) \in R$. Moreover, from both the pre-signature $\tilde{\sigma}$ and the adapted signature σ, the signer can extract a witness of Y. AS can be viewed as an extension of (ordinary) signatures by additionally addressing mutual trust between the signer and the receiver, since the secret witness is exposed to the signer once the full signature has been published.

AS are widely applied in fair exchanges [8], atomic swaps [22, 13], and payment channel networks [2, 13] to reduce on-chain computations and improve the fungibility of transactions. We briefly discuss the applications of AS as follows.

Fair exchange of a witness. Assume Alice, who holds a token c for some cryptocurrency, e.g., some amount of ETH for Ethereum, wants to trade it for a witness y of some instance Y held by Bob (where y may be some secret information accessing some digital services). Alice can post to the blockchain a timeout transaction transferring c to Bob, which however requires a full AS signature (with respect to Y) to be claimed; then, off-chain, Alice can pre-sign a transaction tx using Y and send the pre-signature $\tilde{\sigma}$ to Bob. AS allow then Bob to adapt $\tilde{\sigma}$ to a full signature σ using y, and upload it to the blockchain to receive c. Once σ has been published, Alice can extract the witness y which completes the exchange.

Atomic swaps. Atomic swaps [22, 13] allow two parties, Alice and Bob, to exchange assets in two different cryptocurrencies c_A and c_B. First, both Alice and Bob lock c_A and c_B on the blockchain as deposits (a.k.a. collateral). Then, Alice randomly samples an instance-witness pair (Y, y), generates a pre-signature $\tilde{\sigma}_A$ on message tx_A and instance Y, and sends tx_A, Y, $\tilde{\sigma}_A$ to Bob. Here, tx_A is a transaction for transferring c_A to Bob. Then, Bob also generates a pre-signature $\tilde{\sigma}_B$ on message tx_B and instance Y, and sends tx_B, $\tilde{\sigma}_B$ to Alice, where tx_B is a transaction for transferring c_B to Alice. After receiving $\tilde{\sigma}_B$, Al-

ice can adapt it to a full signature σ_B with the knowledge of y, and upload it to the blockchain to obtain c_B. Meanwhile, Bob can also extract y from $\tilde{\sigma}_B$ and σ_B, adapt $\tilde{\sigma}_A$ to σ_A, and hence obtain c_A.

Multi-hop payments. Multi-hop payments [13] allow multiple parties to route payments between them, provided that they have a payment channel with a common intermediate[4]. Consider four parties Alice, Bob, Charlie and David, where Alice wants to pay cryptocurrency (say c) to David. First, Alice and Bob lock some funds on the blockchain on a payment channel as deposits/collateral, Bob with Charlie, and Charlie with David do likewise. Then, David randomly samples an instance/witness pair (Y, y) and forwards Y to Alice, Bob, and Charlie. Subsequently, Alice generates a pre-signature $\tilde{\sigma}_A$ on message tx_A and instance Y, and then sends $tx_A, \tilde{\sigma}_A$ to Bob. Here tx_A is a transaction for transferring c to Bob. Then Bob also generates a pre-signature $\tilde{\sigma}_B$ on message tx_B (transaction for transferring c to Charlie) and instance Y and sends $tx_B, \tilde{\sigma}_B$ to Charlie. After that, Charlie generates a pre-signature $\tilde{\sigma}_C$ on message tx_C (transaction for transferring c to David) and instance Y, and sends $tx_C, \tilde{\sigma}_C$ to David. After receiving $\tilde{\sigma}_C$, David can adapt it to a full signature σ_C with the knowledge of y, and upload it to the blockchain to obtain c. Meanwhile, Charlie can also extract y from $\tilde{\sigma}_C$ and σ_C, adapt $\tilde{\sigma}_B$ to σ_B and hence obtain c. Finally Bob can follow the same procedure to obtain c.

Security of Adaptor Signatures. The security definition of AS was formalized by Aumayr *et al.* [2, 3] and adopted by almost all subsequent works[5] ([13, 25, 27, 18, 12, 26], to name a few). We give the formal security definition of AS in Section 2.3. Nonetheless, as finding an issue with the existing definition (and repairing it) is one of our contributions, we provide here an informal discussion.

As an extension of signatures, AS inherits the classical unforgeability property of signatures. Namely, only the owner of the secret key can generate a valid pre-signature (and a regular signature, of course). Besides classical unforgeability, two additional properties are required for the security of the sender and the receiver.

[4] In the original protocol in [13], each party is sampling a new pair of instance-witness and they are using it once for each payment. Here we simplify the protocol by allowing every party to take the same instance. The security still holds assuming that the intermediate parties do not collude.

[5] Dai *et al.* [10] identified that Aumayr *et al.*'s definition [2, 3] does not consider the case of multiple pre-sign queries by the adversary, and fixed it by proposing a so-called full extractability property.

Security for the the sender (a.k.a. witness extractability). The sender can extract a witness from the valid pre-signature and the valid adapted signature.

Security for the receiver (a.k.a. pre-signature adaptability). The receiver can adapt a valid pre-signature into a valid (full) signature with the knowledge of a witness.

Notice that an AS scheme is defined with respect to a hard relation R, which can vary from simple discrete logarithm relations to more complex relations based on a blockchain scripting language. Therefore, a natural question is:

What relation R can an adaptor signature scheme support?

Adaptor Signatures for NP relations. Most previous works [2, 27, 13, 25, 18] focus on constructing AS schemes based on particular signatures schemes (like the ECDSA and Schnorr) and for script-related relations (like the public/secret key relation of signatures). The more recent work by Dai, Okamoto, and Yamamoto [10] gave a generic constructions of AS for general NP relations. They showed that AS can be constructed from any signature scheme and any NP-hard relation, and therefore, adaptor signatures are implied by one-way functions.

An advantage of the construction from [10] is its simplicity: Let SIG be a signature scheme, and R be an NP relation. In Dai *et al.*'s generic construction GAS1, a pre-signature of message m w.r.t. instance Y is in the form of $\tilde{\sigma} = (\bar{\sigma}, Y)$, where $\bar{\sigma}$ is a signature of SIG for message (m, Y). To adapt $\tilde{\sigma}$ into a full signature, one just attaches the witness y to $\tilde{\sigma}$ and obtains $\sigma = (\bar{\sigma}, Y, y)$. The verification algorithm of AS checks both the validity of $\bar{\sigma}$ w.r.t. message (m, Y) and that $(Y, y) \in R$.

However, as it turns out the above simplicity comes at a high cost: the witness y is exposed in the adapted signature in plain text. This poses serious security risks in many applications. For example, in the fair exchange application above, the adapted signature σ is uploaded to the blockchain, making the witness accessible to everyone on the network. However, y should only be known to the buyer (Alice) since she has made a payment to the seller (Bob).

Similar security issues also arise in multi-hop payments when the construction by [10] is used. Consider the case that Alice wants to pay to David via two intermediary nodes, Bob and Charlie. If y is contained in plain in an adapted signature, then after David uploads σ_C, Bob is able to get y and adapt $\tilde{\sigma}_A$ into σ_A, thus skipping Charlie and receiving his money, which is conflict with the fairness of multi-hop payments.

As it turns out, the above issue is not just an issue of the construction in [10], but rather a deeper issue with the definition of security of AS. Intuitively, the functionality of AS requires that a witness can be extracted from *both* the pre-signature $\tilde{\sigma}$ and the adapted signature σ, but not from either of them individually. In almost all existing AS schemes [2, 27, 13, 11, 18], both $\tilde{\sigma}$ and σ are essential for extracting a witness. However, the generic construction GAS1 from [10] satisfies all security requirements of the formal AS security definition by Aumayr *et al.* [2, 3]—including unforgeability, witness extractability, and pre-signature adaptability—but still fails in satisfying the above intuition. This demonstrates that previous security definition does not cover all security properties needed for the applications of AS, and points to a new hole in the literature (which we fill in) of a generic AS construction for any NP relation.

To solve the first (definitional) problem, we introduce a new security property called *witness hiding*. Informally, it requires that the witness y can be extracted from both a pre-signature and an adapted signature (jointly), but not from only one of them alone. Thus the key open question now is:

Question: *Can witness hiding adaptor signatures support any NP relation, and, if so, what is the minimal assumption for such a construction?*

In this work, we propose a generic construction of AS from any signature scheme and for any NP relation. Since signatures can be constructed from one-way functions [15], we obtain the following theorem.

Theorem 1. *Assuming one-way functions exist, then there exist witness hiding adaptor signatures for any NP language.*

We summarize our contributions as follows:

– We introduce witness hiding, a new security property for adaptor signatures. This property requires that the witness y can be extracted from both a pre-signature and an adapted signature, but not from only one of them. Witness hiding is crucial in most applications of AS, including fair exchanges [8], atomic swaps [22, 13], and payment channel networks [2, 13], where the pre-signature remains private while the adapted signature is uploaded to the blockchain and is public to everyone. Witness hiding helps prevent an eavesdropper from extracting a witness from only the adapted signature. We observe that the only

existing adaptor signature scheme for any NP relation [10] does not satisfy witness hiding, as the witness is exposed in plain in the adapted signature[6].

– We propose a generic construction of witness-hiding adaptor signatures from (ordinary) signatures and a new type of trapdor commitment which we term *trapdoor commitments with a specific adaptable message.* The latter is a weaker version of classical trapdoor commitments, where there is a specific message $m_{\mathbf{0}}$, and with the knowledge of the trapdoor, one can open a commitment of $m_{\mathbf{0}}$ to another message m.
– We propose a trapdoor commitment scheme with a specific adaptable message based on the Hamiltonian cycle problem, where the commitment key is the Hamiltonian problem instance and the trapdoor is a Hamiltonian cycle (witness). Since the Hamiltonian cycle problem is NP-complete, we obtain adaptor signatures for any NP language. See Fig. 1 for a framework of adaptor signatures.

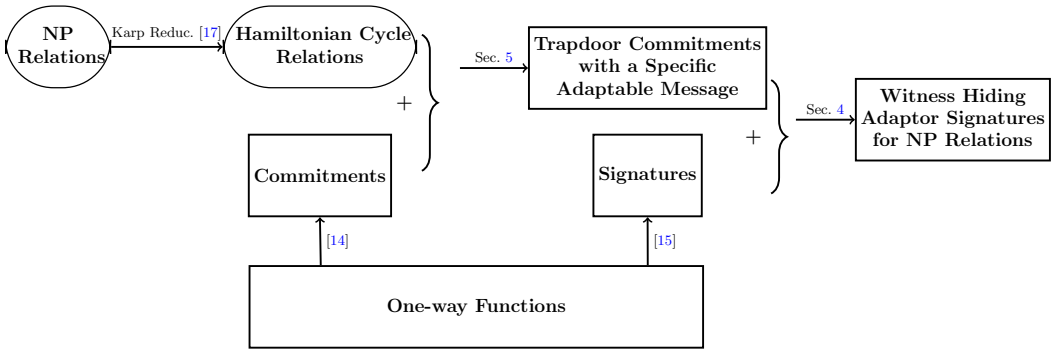

Fig. 1. A framework of witness hiding adaptor signatures for NP relations

1.1 Related Work

The concept of adaptor signatures (AS) was introduced by Poelstra [21, 22] (referred as scriptless scripts in [21]). In 2020, Aumayr *et al.* [2, 3] first formalized adaptor signatures, and proposed three security properties for AS: unforgeability, pre-signature adaptability, and witness extractability.

[6] Almost all previous constructions ([2, 27, 13, 25, 18, 11]), except GAS1 [10] satisfy witness hiding property.

All subsequent follow-up works on AS can be categorized into two main directions.

The first direction focuses on designing AS from known underlying signature schemes. For example, the ECDSA-based adaptor signature scheme [2], the Schnorr-based scheme [2, 27], the LWE/SIS-based scheme LAS [13], the isogeny-based scheme IAS [25], the code-based scheme [18], etc. Note that the supported languages (e.g., the discrete logarithm language) in these schemes are fixed, due to the specific structures of the underlying signature schemes.

The second direction focuses on generic constructions of AS [11, 10]. Erwig *et al.* [11] showed that identification (ID) schemes with additional homomorphic properties can be transformed into adaptor signatures. However, the transform requires the supporting language to be highly related with the format of the commitment in the ID scheme, limiting the instantiations to the DL-based or the RSA-based ID schemes and their corresponding languages.

Dai *et al.* [10] proposed the first truly generic construction of AS, called GAS1, from any signature scheme and any NP language. As discussed however the (overly-)simple construction GAS1, despite satisfying the security requirements according to [2], has the significant issue: that the witness y is exposed in the adapted signature, which poses a serious security vulnerability in blockchain applications.

In fact, [10] also proposed a second generic construction called GAS2 from any signature scheme and any strongly random-self reducible relation. Compared to GAS1, GAS2 is unlinkable, i.e., the adapted signature is indistinguishable from a normally generated signature, and hence the witness is hidden from only the adapted one. However, GAS2 requires the strong random-self reducibility of the underlying relation. Therefore, similar to [11], the instantiations of GAS2 are limited to standard number theoretical problems such as DL, RSA, and LWE.

1.2 Technical Overview

In this subsection, we provide a brief overview of our techniques.

Defining Witness Hiding. The witness hiding property requires that the witness y is extractable from both the pre-signature $\tilde{\sigma}$ and the adapted signature σ (jointly), but not from either of them alone. Note that y is inherently hidden in $\tilde{\sigma}$ since the pre-sign algorithms takes only the message, the instance and the secret key as inputs, and it is independent of y. More formally, we say an AS scheme has the witness hiding property,

if there exists an simulator that, given the secret key (of AS), the message, and the instance as inputs, outputs a signature which is indistinguishable from a signature adapted from a pre-signature using witness y.

Generic Construction. Next we illustrate our generic construction of AS from any signature scheme and for any NP relation. Our approach draws inspiration from the simplicity of [10] but significantly modifies their paradigm with novel ideas to achieve the witness-hiding property.

More concretely, let SIG be an ordinary signature scheme and R be a hard relation. Recall that in GAS1, a pre-signature for message m and instance Y is in the form of $\tilde{\sigma} = (\bar{\sigma}, Y)$, where $\bar{\sigma}$ is a signature of SIG for message (m, Y). And the adapted signature is just $\sigma = (\bar{\sigma}, Y, y)$ by attaching the witness to $\tilde{\sigma}$.

One might be tempted to use the following idea to avoid the exposure of y in σ: replace y with a zero-knowledge proof, and show that the adaptor knows a witness of Y. However, in such a modification, y cannot be extracted from both $\tilde{\sigma}$ and σ, and the witness extractability is violated.

Our key insight here is the observation that the witness extractability property of AS has similarities in spirit with the special soundness property of Sigma protocols. To demonstrate this, let us first recall this property. A Sigma protocol for a hard relation R is a three-move protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, where the prover holds a witness y of some instance Y and wants to prove to the verifier in a zero-knowledge way. A complete transcript of a protocol execution consists of three parts: the first move is a commitment cmt sent from $\mathcal{P}$ to $\mathcal{V}$; the second move is a random challenge ch from $\mathcal{V}$ to $\mathcal{P}$; and the third move is a response rsp from $\mathcal{P}$ to $\mathcal{V}$.

- **Special soundness of Sigma protocols.** From two valid transcripts with the same commitment but different challenges, one can extract a witness of the instance.
- **Witness extractability of AS.** From a valid pre-signature and an adapted signature one can extract a witness.

Inspired by this analogy observation, we modify the pre-signature of AS to be $\tilde{\sigma} = (\bar{\sigma}, Y, (cmt, ch', rsp'))$, where $\bar{\sigma}$ is a signature of (m, Y, cmt), and (cmt, ch', rsp') is a valid transcript of a Sigma protocol w.r.t. instance Y. To adapt it to a (full) signature, the adaptor, with the knowledge of y, has to generate rsp for another challenge $ch \neq ch'$ such that (cmt, ch, rsp) is also a valid transcript. This is feasible due to the completeness of the Sigma protocol, since a prover knowing a witness y is able

to answer any challenge and reply a response to make the transcript valid. Meanwhile, the witness extractability of AS is guaranteed by the special soundness of the Sigma protocol, since from $\tilde{\sigma} = (\bar{\sigma}, Y, (cmt, ch, rsp))$ and $\sigma = (\bar{\sigma}, Y, (cmt, ch' \neq ch, rsp'))$ one can extract a witness of Y.

We formally describe our generic construction using trapdoor commitments (TC, a.k.a. chameleon hashes [19], which are equivalent to Sigma protocols [4]). In a TC scheme, with the public commitment key, one can commit a message m' to get a commitment c and an opening d', and with the trapdoor one can open c to another $m \neq m'$ and get a corresponding d. Meanwhile, trapdoor extractability requires that from a collision (c, m', d') and $(c, m \neq m', d)$ one can extract the trapdoor. In our generic construction of AS, we first transfer the instance-witness pair (Y, y) into a commitment-trapdoor key pair of a TC scheme. Now, a pre-signature w.r.t. message m and instance Y is in the form of

$$\tilde{\sigma} = (\bar{\sigma}, Y, c, m' \neq m, d'),$$

where $\bar{\sigma}$ is a signature for (m, Y, c), and d' is an opening of c for a "dummy" message $m' \neq m$. Given $\tilde{\sigma}$ and witness y (the trapdoor of the underlying TC scheme), the adapted signature is in the form of

$$\sigma = (\bar{\sigma}, Y, c, m, d),$$

where d is another opening for the signed message m, adapted from (c, m', d') using trapdoor (witness) y.

Functionality and security of the generic construction are analyzed as follows.

- **Functionality of adaption.** This is guaranteed by the trapdoor adaption property of TC.
- **Unforgeability.** This is inherited from the underlying signature scheme.
- **Witness extractability.** This is guaranteed by the trapdoor extractability of TC. Namely, from a collision (c, m', d') and $(c, m \neq m', d)$ one can extract the trapdoor.
- **Pre-signature adaptability.** This is guaranteed by the functionality of TC, i.e., with the trapdoor one can open a commitment to any message, and therefore the adapted signature $\sigma = (\bar{\sigma}, Y, c, m, d)$ is valid.
- **Witness hiding.** This is due to the fact that from only a tuple of commitment (c, m, d) nothing about the trapdoor is leaked.

We notice that in the pre-sign process, the dummy message m' in the pre-signature can be a fixed value $m_{\mathbf{0}}$, as long as it differs from the

message m to be signed. Moreover, to construct AS schemes, we only require a property that a commitment of the fixed $m_{\mathbf{0}}$ (but not necessarily an arbitrary commitment) can be opened to another message. Based on this observation, we propose a weakened notation of trapdoor commitments, termed trapdoor commitments with a specific adaptable message, where there exists a specific message $m_{\mathbf{0}}$, and with the trapdoor one can (only) open a commitment of $m_{\mathbf{0}}$ to another message. Next we will see, such a weakening enables constructions from any NP relation, where the instance Y and the witness y serve as the commitment key and the trapdoor, respectively.

Constructing TC with a Specific Adaptable Message. We now turn to construct a trapdoor commitment scheme with a specific adaptable message for any NP relation R. Bellare and Ristov [4] proved the equivalence of Sigma protocols and chameleon hashes (and hence trapdoor commitments), where the commitment, the challenge, and the response in a Sigma protocol correspond to the commitment, the message, and the opening in a trapdoor commitment scheme. However, one must exercise caution when transferring one to another, since their security definitions are not perfectly matched. For example, to transfer a Sigma protocol into a trapdoor commitment scheme, we must additionally ensure that there is a simulator for the Sigma protocol, which can generate a simulated transcript given a fixed challenge, and the commitment can be recovered from the challenge, the response, and the instance. However, this is not a universally applicable property for all Sigma protocols.

Following the framework by Bellare and Ristov [4], we found that the Sigma (zero-knowledge) protocol for the Hamiltonian cycle problem by Blum [7, 14] can be perfectly transferred into a trapdoor commitment scheme with a specific adaptable message $m_{\mathbf{0}} = 0$. We recall the protocol and present the corresponding trapdoor commitment scheme in Fig. 2.

Let G be a graph, and $H \subseteq G$ be a Hamiltonian cycle, i.e., a witness of Hamiltonian graph instance G. First, the prover $\mathcal{P}$ randomly samples a permutation π and commits $G' = \pi(G)$, and then sends the commitments $com_{G'}$[7] to the verifier $\mathcal{V}$. Here com denotes standard commitments with statistical biding and computational hiding (cf. Definition 1). Then the verifier $\mathcal{V}$ sends a random challenge $ch \xleftarrow{\$} \{0, 1\}$. If $ch = 0$, then $\mathcal{P}$ sends all openings of $com_{G'}$ and the permutation π to $\mathcal{V}$, and $\mathcal{V}$ checks $com_{G'}$ are commitments of $\pi(G)$. And if $ch = 1$, then $\mathcal{P}$ sends all openings of

[7] More precisely, $com_{G'}$ is a group of bit commitments for the adjacency matrix of G'.

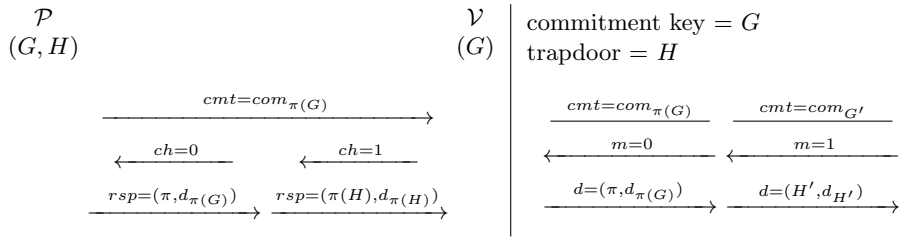

Fig. 2. The zero-knowledge proof protocol for the Hamiltonian cycle problem [7, 14] (left) and the trapdoor commitment scheme from it (right). Here $(G, H \subseteq G)$ is an instance-witness pair of the Hamiltonian cycle problem, and *com* and d are commitments and openings of a bit commitment scheme with statistical biding and computational hiding.

$com_{H'}$ to $\mathcal{V}$, and $\mathcal{V}$ checks $com_{G'}$ include commitments of a Hamiltonian cycle $H'(= \pi(H))$.

The Sigma protocol described above has a zero-knowledge simulator that, given the challenge ch, can perfectly simulate a transcript (cmt, ch, rsp). Moreover, if $ch = 0$, then the simulated transcript is identical to the transcript from an honest execution. And with the knowledge of a witness, it is easy to get a response for $ch = 1$ under the same commitment, which is exactly the functionality of adaption in the trapdoor commitment scheme.

Our trapdoor commitment scheme with specific adaptable message $m_{\mathbf{0}} = 0$ is shown in Fig. 2 (right). If $m = 0$, then the commitment is $com_{\pi(G)}$ and the corresponding opening is $(\pi, d_{\pi(G)})$, where π is a random permutation and $d_{\pi(G)}$ is the corresponding openings of the underlying (standard) commitment scheme. If $m = 1$, then the commitment is $com_{G'}$ and the corresponding opening is $d_{H'}$, where G' is a randomly generated Hamiltonian graph with a Hamiltonian cycle H'.

Given that the Hamiltonian cycle problem is NP-complete [17], we know any NP relation R can be transferred into a trapdoor commitment scheme with a specific adaptable message. Therefore, we get witness hiding adaptor signature schemes from any signature scheme and for any NP relation. Combined with the fact that signature schemes and (standard) bit commitment schemes are implied by one-way functions [14, 15], Theorem 1 holds consequently.

1.3 Organization of the Paper

This rest of the paper is organized as follows. In Section 2, we present preliminaries and define trapdoor commitments with a specific adaptable message. In Section 3, we introduce adaptor signatures and their security

properties, including the witness hiding property. Section 4 details the generic construction of AS. The trapdoor commitment scheme with a specific adaptable message for the Hamiltonian cycle problem is shown in Section 5. Finally, we conclude this paper in Section 6.

2 Preliminaries

Throughout this paper, we use $\lambda \in \mathbb{N}$ to denote the security parameter. For $\mu \in \mathbb{N}$, define $[\mu] := \{1, 2, ..., \mu\}$. Denote by $x := y$ the operation of assigning y to x. Denote by $x \xleftarrow{\$} \mathcal{S}$ the operation of sampling x uniformly at random from a set $\mathcal{S}$. For a distribution $\mathcal{D}$, denote by $x \leftarrow \mathcal{D}$ the operation of sampling x according to $\mathcal{D}$. For an algorithm $\mathcal{A}$, denote by $y \leftarrow \mathcal{A}(x; r)$, or simply $y \leftarrow \mathcal{A}(x)$, the operation of running $\mathcal{A}$ with input x and randomness r and assigning the output to y. For deterministic algorithms $\mathcal{A}$, we also write as $y := \mathcal{A}(x)$ or $y := \mathcal{A}(x; r)$. "PPT" is short for probabilistic polynomial-time.

2.1 Commitments

Definition 1 (Commitments). *A commitment scheme consists of the following three algorithms. Namely,* $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Ver})$.

- $ck \leftarrow \mathsf{Gen}(1^\lambda)$. *The key generation algorithm takes as input the security parameter* λ*, and outputs a commitment key* ck.
- $(c, d) \leftarrow \mathsf{Com}(ck, m)$. *The commitment algorithm takes as input* ck *and a message* $m \in \mathcal{M}$*, and outputs a commitment* c *and an opening* d.
- $0/1 \leftarrow \mathsf{Ver}(ck, c, m, d)$. *The verification algorithm takes as input* ck*,* c*,* m *and* d*, and outputs a bit.*

Correctness. *For any* $ck \leftarrow \mathsf{Gen}(1^\lambda)$*, any message* $m \in \mathcal{M}$ *and* $(c, d) \leftarrow \mathsf{Com}(ck, m)$*, it holds that* $\mathsf{Ver}(ck, c, m, d) = 1$.

Definition 2 (Statistical Biding of Commitments). *A commitment scheme* COM *has statistical biding, if for any unbounded adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{biding}_{\mathsf{COM},\mathcal{A}}(\lambda) := \left| \Pr \left[\begin{array}{c} ck \leftarrow \mathsf{Gen}(1^\lambda) \\ (c, m_0, m_1, d_0, d_1) \leftarrow \mathcal{A}(ck) \end{array} : \begin{array}{c} m_0 \neq m_1 \\ \wedge\ \mathsf{Ver}(ck, c, m_0, d_0) = 1 \\ \wedge\ \mathsf{Ver}(ck, c, m_1, d_1) = 1 \end{array} \right] \right|$$

is negligible over λ.

Definition 3 (Hiding of Commitments). *A commitment scheme* COM *has hiding, if for any PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{hiding}_{\mathsf{COM},\mathcal{A}}(\lambda) := \left| \Pr\left[\begin{matrix} ck \leftarrow \mathsf{Gen}(1^\lambda); (m_0, m_1, st) \leftarrow \mathcal{A}(ck) \\ (c, d) \leftarrow \mathsf{Com}(ck, m_0) \end{matrix} : \mathcal{A}(st, c) = 1 \right] \right.$$
$$\left. - \Pr\left[\begin{matrix} ck \leftarrow \mathsf{Gen}(1^\lambda); (m_0, m_1, st) \leftarrow \mathcal{A}(ck) \\ (c, d) \leftarrow \mathsf{Com}(ck, m_1) \end{matrix} : \mathcal{A}(st, c) = 1 \right] \right|$$

is negligible over λ*.*

Bit commitment schemes (i.e., the message is one bit) with statistical biding can be constructed from one-way functions [14].

2.2 Trapdoor Commitments

Definition 4 (Trapdoor Commitments with Specific Adaptable Message). *Let* $\mathcal{M}$ *be a message space and* $m_{\mathbf{0}} \in \mathcal{M}$*. A trapdoor commitment (TC) scheme with specific adaptable message* $m_{\mathbf{0}}$ *consists of the following four algorithms. Namely,* $\mathsf{TC} = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Ver}, \mathsf{TdOpen})$*.*

- $(ck, td) \leftarrow \mathsf{Gen}(1^\lambda)$*. The key generation algorithm takes as input the security parameter* λ*, and outputs a commitment key* ck *and a trapdoor* td*.*
  We implicitly assume ck *is contained in* td*, and there exists an efficient function to check the validity of a trapdoor w.r.t. a commitment key, i.e.,* $f_{\mathsf{TC}}(ck, td) = 1$ *if* td *is valid w.r.t.* ck*.*
- $(c, d) \leftarrow \mathsf{Com}(ck, m)$*. The commitment algorithm takes as input* ck *and a message* $m \in \mathcal{M}$*, and outputs a commitment* c *and an opening* d*.*
- $0/1 \leftarrow \mathsf{Ver}(ck, c, m, d)$*. The verification algorithm takes as input* ck*,* c*,* m *and* d*, and outputs a bit.*
- $d \leftarrow \mathsf{TdOpen}(td, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m)$*. The trapdoor open algorithm takes as input* td*,* c*,* $m_{\mathbf{0}}$*,* $d_{\mathbf{0}}$*, and another message* m*, and outputs an adapted opening* d*.*

Correctness*. For any* $(ck, td) \leftarrow \mathsf{Gen}(1^\lambda)$*, any message* $m \in \mathcal{M}$*, the followings two hold.*

1. *If* $(c, d) \leftarrow \mathsf{Com}(ck, m)$*, then* $\mathsf{Ver}(ck, c, m, d) = 1$*.*
2. *If* $(c, d_{\mathbf{0}}) \leftarrow \mathsf{Com}(ck, m_{\mathbf{0}})$ *and* $d \leftarrow \mathsf{TdOpen}(td, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m)$*, then* $\mathsf{Ver}(ck, c, m, d) = 1$*.*

Definition 5 (Hiding of Trapdoor Commitments). *A trapdoor commitment scheme* TC *with specific adaptable message* $m_{\mathbf{0}}$ *has hiding, if for any PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{hiding}_{\mathsf{TC},\mathcal{A}}(\lambda) := \left| \Pr\left[\begin{matrix} (ck, td) \leftarrow \mathsf{Gen}(1^\lambda); (m, st) \leftarrow \mathcal{A}(ck) \\ (c, d_{\mathbf{0}}) \leftarrow \mathsf{Com}(ck, m_{\mathbf{0}}) \end{matrix} : \mathcal{A}(st, c) = 1 \right] \right.$$
$$\left. - \Pr\left[\begin{matrix} (ck, td) \leftarrow \mathsf{Gen}(1^\lambda); (m, st) \leftarrow \mathcal{A}(ck) \\ (c, d) \leftarrow \mathsf{Com}(ck, m) \end{matrix} : \mathcal{A}(st, c) = 1 \right] \right|$$

is negligible over λ*.*

Definition 6 (Trapdoor Extractability of Trapdoor Commitments). *A trapdoor commitment scheme* TC *with specific adaptable message* $m_{\mathbf{0}}$ *is trapdoor extractable, if there is an efficient extract algorithm* Ext *that can extract a trapdoor from a collision with high probability. More precisely, for* $(ck, td) \leftarrow \mathsf{Gen}(1^\lambda)$*, any* $(c, m_{\mathbf{0}}, d_{\mathbf{0}})$ *and* (c, m, d) *s.t.* $m \neq m_{\mathbf{0}}$ *and* $\mathsf{Ver}(ck, c, m_{\mathbf{0}}, d_{\mathbf{0}}) = \mathsf{Ver}(ck, c, m, d) = 1$*, it holds that*

$$\Pr[f_{\mathsf{TC}}(ck, \mathsf{Ext}(ck, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m, d)) = 0] \leq negl(\lambda),$$

where the probability is taken over the random choice of key generation.

Definition 7 (Adaption Indistinguishability of Trapdoor Commitments). *A trapdoor commitment scheme* TC *with a specific adaptable message* $m_{\mathbf{0}}$ *has adaption indistinguishability, if for any* m*, any PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{aind}_{\mathsf{TC},\mathcal{A}}(\lambda) := \left| \Pr\left[\begin{matrix} (ck, td) \leftarrow \mathsf{Gen}(1^\lambda) \\ (c, d_{\mathbf{0}}) \leftarrow \mathsf{Com}(ck, m_{\mathbf{0}}) \\ d \leftarrow \mathsf{TdOpen}(td, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m) \end{matrix} : \mathcal{A}(ck, m_{\mathbf{0}}, m, c, d) = 1 \right] \right.$$
$$\left. - \Pr\left[\begin{matrix} (ck, td) \leftarrow \mathsf{Gen}(1^\lambda) \\ (c, d) \leftarrow \mathsf{Com}(ck, m) \end{matrix} : \mathcal{A}(ck, m_{\mathbf{0}}, m, c, d) = 1 \right] \right|$$

is negligible over λ*.*

Remark 1 (Classical Trapdoor Commitments). If $m_{\mathbf{0}}$ in the above definitions is replaced with an arbitrary message m, then we define the classical trapdoor commitment schemes, and the corresponding properties of hiding, trapdoor extractability, and adaption indistinguishability. Jumping ahead, the commitment c of the specific adaptable message $m_{\mathbf{0}}$ serves as one part of the message to be signed in the pre-sign process of adaptor signatures. With the knowledge of the witness y, which is the trapdoor in the commitment scheme, one can open this commitment c to the real message to be signed and hence form a valid adapted signature.

2.3 Signatures

Definition 8 (Signatures). *A signature scheme consists of the following three algorithms. Namely,* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$.

- $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$. *The key generation algorithm takes as input the security parameter* λ*, and outputs a public key* pk *and a secret key* sk.
- $\sigma \leftarrow \mathsf{Sign}(sk, m)$. *The signing algorithm takes as input* sk *and a message* m*, and outputs a signature* σ.
- $0/1 \leftarrow \mathsf{Ver}(pk, m, \sigma)$. *The verification algorithm takes as input* pk*,* m*, and* σ*, and outputs a bit* b *indicating the validity of* σ *(w.r.t.* m*).*

Correctness. *For any* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$*, any message* m *and* $\sigma \leftarrow \mathsf{Sign}(sk, m)$*, it holds that* $\mathsf{Ver}(pk, m, \sigma) = 1$.

Definition 9 (Unforgeability of Signatures). *A signature scheme* SIG *is unforgeable under chosen message attacks (UF-CMA secure), if for any PPT adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{uf}_{\mathsf{SIG},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{uf}_{\mathsf{SIG},\mathcal{A}}(\lambda) \Rightarrow 1]$ *is negligible over* λ*, where* $\mathsf{Exp}^{uf}_{\mathsf{SIG},\mathcal{A}}(\lambda)$ *is defined in Fig. 3.*

$\mathsf{Exp}^{uf}_{\mathsf{SIG},\mathcal{A}}(\lambda)$:	$\textsc{Sign}(m)$:
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$; $\mathcal{S} := \varnothing$	$\sigma \leftarrow \mathsf{Sign}(sk, m)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdot)}(pk)$	$\mathcal{S} := \mathcal{S} \cup \{m\}$
Return $((m^* \notin \mathcal{S}) \wedge (\mathsf{Ver}(pk, m^*, \sigma^*)))$	Return σ

Fig. 3. The UF-CMA security experiment of signatures

2.4 NP Languages

Let $\{R_\lambda\} \subseteq (\{0,1\}^* \times \{0,1\}^*)_\lambda$ be a series of binary relations indexed by parameter λ. If λ is fixed then we simply denote R_λ as R. We call R an NP relation if there is an efficient algorithm to check whether $(Y, y) \in R$. The relation R defines an NP language $\mathcal{L}_R := \{Y \in \{0,1\}^* \mid \exists y \in \{0,1\}^* \; s.t. \; (Y, y) \in R\}$. We call Y the instance (not necessarily in $\mathcal{L}_R$), and y a witness of Y if $(Y, y) \in R$. Usually, there is an efficient sample algorithm that returns an instance-witness pair. Formally, $(Y, y) \leftarrow \mathsf{Sample}(R)$.

Definition 10 (Hard Relations). *A binary relation* R *is hard (one-way) if for any PPT adversary* $\mathcal{A}$*, its advantage*

$$\mathsf{Adv}^{ow}_{R,\mathcal{A}}(\lambda) := \Pr[(Y, y) \leftarrow \mathsf{Sample}(R); y' \leftarrow \mathcal{A}(R, Y) \;:\; (Y, y') \in R]$$

is negligible over λ.

3 Adaptor Signatures

In this section, we present the definition of adaptor signatures and the security requirements, including the newly proposed witness hiding property.

Definition 11 (Adaptor Signatures). *An adaptor signature scheme w.r.t. a relation R consists of seven algorithms $\mathsf{AS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{pSign}, \mathsf{pVer}, \mathsf{Adapt}, \mathsf{Ext})$, where the first three algorithms are defined as regular signatures (cf. Def. 8), and the last four are defined as follows.*

- $\tilde{\sigma} \leftarrow \mathsf{pSign}(sk, m, Y)$. *The pre-sign algorithm takes as input sk, m and an instance Y, and outputs a pre-signature $\tilde{\sigma}$.*
- $b \leftarrow \mathsf{pVer}(pk, m, Y, \tilde{\sigma})$. *The pre-verification algorithm takes as input pk, m, Y and $\tilde{\sigma}$, and outputs a bit indicating the validity of $\tilde{\sigma}$.*
- $\sigma \leftarrow \mathsf{Adapt}(pk, m, \tilde{\sigma}, y)$. *The adaption algorithm takes as input pk, m, $\tilde{\sigma}$ and a witness y as input, and outputs an adapted signature σ.*
- $y/\perp \leftarrow \mathsf{Ext}(pk, m, Y, \tilde{\sigma}, \sigma)$. *The extraction algorithm takes as input pk, m, Y, $\tilde{\sigma}$ and σ, and outputs a witness y, or a failure symbol $\perp$.*

Except the correctness as defined in Def. 8, we additionally require the pre-signature correctness and extraction correctness.

Pre-signature correctness and extraction correctness. *For any $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, any message m, any $(Y, y) \in \mathcal{R}$, $\tilde{\sigma} \leftarrow \mathsf{pSign}(sk, m, Y)$, and $\sigma \leftarrow \mathsf{Adapt}(pk, m, \tilde{\sigma}, y)$, it holds that*

1. *(Pre-signature correctness) $\mathsf{pVer}(pk, m, Y, \tilde{\sigma}) = 1$, and*
2. *(Extraction correctness) $\mathsf{Ver}(pk, m, \sigma) = 1$.*

We require unforgeability, witness extractability and pre-signature adaptability for the security of adaptor signatures. Here, we mainly follow the definition by Dai *et al.* [10] which allows multiple queries to the pre-sign oracle. Meanwhile, we divide the full extractability in [10] into unforgeability and witness extractability (as [2]) for better presenting the different security aspects of adaptor signatures.

Definition 12 (Unforgeability of Adaptor Signatures). *An adaptor signature scheme AS w.r.t. binary relation R is unforgeable under chosen message attacks (UF-CMA secure), if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{uf}_{\mathsf{AS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{uf}_{\mathsf{AS},\mathcal{A}}(\lambda) \Rightarrow 1]$ is negligible over λ, where $\mathsf{Exp}^{uf}_{\mathsf{AS},\mathcal{A}}(\lambda)$ is defined in Fig. 4.*

$\mathsf{Exp}^{uf}_{\mathsf{AS},\mathcal{A}}(\lambda)$:	$\underline{\textsc{Sign}(m)}$:
$\underline{(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)}$; $\mathcal{S} := \varnothing$; $\mathcal{T}[m] := \varnothing$	$\sigma \leftarrow \mathsf{Sign}(sk, m)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdot),\textsc{pSign}(\cdot,\cdot),\textsc{NewY}()}(pk)$	$\mathcal{S} := \mathcal{S} \cup \{m\}$
	Return σ
Return 1 if $(b_1 \wedge (b_{2,1} \vee b_{2,2}))$, and 0 otherwise, where	$\underline{\textsc{pSign}(m, Y)}$:
b_1: $\mathsf{Ver}(pk, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{S}$	$\tilde{\sigma} \leftarrow \mathsf{pSign}(sk, m, Y)$
$b_{2,1}$: $\mathcal{T}[m^*] = \varnothing$	$\mathcal{T}[m] := \mathcal{T} \cup \{(Y, \tilde{\sigma})\}$
$b_{2,2}$: $\forall (Y, \tilde{\sigma}) \in \mathcal{T}[m^*] : Y \in \mathcal{Y}$	Return $\tilde{\sigma}$
	$\underline{\textsc{NewY}()}$:
	$\underline{(Y, y) \leftarrow \mathsf{Sample}(R)}$
	$\mathcal{Y} := \mathcal{Y} \cup \{Y\}$
	Return Y

Fig. 4. The UF-CMA security experiment of adaptor signatures

Definition 13 (Witness Extractability). *An adaptor signature scheme* AS *w.r.t. relation* R *is witness extractable, if for any PPT adversary* $\mathcal{A}$, $\mathsf{Adv}^{we}_{\mathsf{AS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{we}_{\mathsf{AS},\mathcal{A}}(\lambda) \Rightarrow 1]$ *is negligible over* λ, *where* $\mathsf{Exp}^{we}_{\mathsf{AS},\mathcal{A}}(\lambda)$ *is defined in Fig. 5.*

$\mathsf{Exp}^{we}_{\mathsf{AS},\mathcal{A}}(\lambda)$:	$\underline{\textsc{Sign}(m)}$:
$\underline{(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)}$; $\mathcal{S} := \varnothing$; $\mathcal{T}[m] := \varnothing$	$\sigma \leftarrow \mathsf{Sign}(sk, m)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdot),\textsc{pSign}(\cdot,\cdot)}(pk)$	$\mathcal{S} := \mathcal{S} \cup \{m\}$
	Return σ
Return 1 if $(b_1 \wedge b_2)$, and 0 otherwise, where	$\underline{\textsc{pSign}(m, Y)}$:
b_1: $\mathsf{Ver}(pk, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{S}$	$\tilde{\sigma} \leftarrow \mathsf{pSign}(sk, m, Y)$
b_2: $\forall (Y, \tilde{\sigma}) \in \mathcal{T}[m^*] : (Y, \mathsf{Ext}(pk, m^*, Y, \tilde{\sigma}, \sigma^*)) \notin R$	$\mathcal{T}[m] := \mathcal{T} \cup \{(Y, \tilde{\sigma})\}$
// all $(Y, \tilde{\sigma})$ in the pre-sign list lead to a failed extraction	Return $\tilde{\sigma}$

Fig. 5. The witness extractability experiment of adaptor signatures

Remark 2 (Stronger Definition of Witness Extractability). A stronger definition of witness extractability is that the extraction will not fail as long as $\mathsf{pVer}(pk, m, Y, \tilde{\sigma}) = 1$. Combined with the correctness, the difference between this stronger definition and Def. 13 lies in whether one can extract a witness from a valid pre-signarture $\tilde{\sigma}$ and an adapted signature σ, such that $\tilde{\sigma}$ is not generated via pSign.

However, such a stronger definition is not practical in the real world, since witness extractability is meant to guarantee Alice's right to extract

y once Bob publishes the adapted signature, assuming Alice generates the pre-signature via (normal) pSign, but not other ways.

Definition 14 (Pre-signature Adaptability). *An adaptor signature scheme* AS *w.r.t. relation* R *has pre-signature adaptability, if for any public key* pk*, any message* m*, any* $(Y, y) \in R$ *and pre-signature* $\tilde{\sigma}$ *s.t.* $\mathsf{pVer}(pk, m, Y, \tilde{\sigma}) = 1$*, it holds that* $\mathsf{Ver}(pk, m, \mathsf{Adapt}(pk, m, \tilde{\sigma}, y)) = 1$.

Now we formally define the property that y can be extracted from both the pre-signature $\tilde{\sigma}$ and the adapted signature σ, but not just σ. In other words, σ leaks no additional information about y.

Definition 15 (Witness Hiding of Adaptor Signatures). *An adaptor signature scheme* AS *w.r.t. relation* R *is witness hiding, if there exists a simulator* Sim *such that, for any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{wh}_{\mathsf{AS},\mathsf{Sim},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{wh}_{\mathsf{AS},\mathsf{Sim},\mathcal{A},0}(\lambda) \Rightarrow 1] - \Pr[\mathsf{Exp}^{wh}_{\mathsf{AS},\mathsf{Sim},\mathcal{A},1}(\lambda) \Rightarrow 1]|$$

is negligible over λ*, where* $\mathsf{Exp}^{wh}_{\mathsf{AS},\mathsf{Sim},\mathcal{A},b}(\lambda)$ *(*$b \in \{0,1\}$*) are defined in Fig. 6.*

$\underline{\mathsf{Exp}^{wh}_{\mathsf{AS},\mathsf{Sim},\mathcal{A},b}(\lambda)}$:	$\underline{\textsc{Chall}_0(pk, sk, m)}$	$\underline{\textsc{Chall}_1(pk, sk, m)}$
$(Y, y) \leftarrow \mathsf{Sample}(R)$	If $f_{\mathsf{AS}}(pk, sk) \neq 1$: Return $\perp$	If $f_{\mathsf{AS}}(pk, sk) \neq 1$: Return $\perp$
Return $\mathcal{A}^{\textsc{Chall}_b(\cdot,\cdot,\cdot)}(Y)$	// check the validity of (pk, sk)	// check the validity of (pk, sk)
	$\tilde{\sigma} \leftarrow \mathsf{pSign}(sk, m, Y)$	$\sigma \leftarrow \mathsf{Sim}(pk, sk, m, Y)$
	$\sigma \leftarrow \mathsf{Adapt}(pk, m, \tilde{\sigma}, y)$	Return σ
	Return σ	

Fig. 6. The witness hiding experiments of adaptor signatures

Remark 3 (On the Formalization of Witness Hiding). Witness hiding property requires that the witness is exposed from both the pre-signature $\tilde{\sigma}$ and the adapted signature σ, but not from either of them. One might wonder why we only ask the adapted signature σ to leak no information about the witness in the above definition, but do not impose restrictions on the pre-signature $\tilde{\sigma}$. Actually, the witness hiding property for $\tilde{\sigma}$ is naturally established, since the pre-sign algorithm takes only the secret key sk, the message m, and the instance Y as input, and hence it is independent of y.

The motivation of introducting witness hiding is to prevent a malicious eavesdropper who has access to blockchains from extracting the witness

using an adapted signature. From this point, the eavesdropper $\mathcal{A}_{eav}$ has no knowledge of the (signing) secret key, and it should be unable to distinguish a pre-sign-and-adapt signature (using y) and a simulated signature. This can be formalized by an experiment where $\mathcal{A}_{eav}$ has access to oracles $\textsc{Chall}_0$ (the pre-sign-and-adaption oracle) and $\textsc{Chall}_1$ (the simulation oracle), and the secret key is sampled by the challenger/experiment and not known to $\mathcal{A}_{eav}$. In the definition above, we consider a stronger adversary $\mathcal{A}$ who is able to select the signing secret key by itself, and $\mathcal{A}$ degrades into an eavesdropper adversary $\mathcal{A}_{eav}$ if it generates sk honestly and discards it immediately after the query. Therefore, Definition 15 is stronger than the definition which considers just the eavesdropping case.

Remark 4 (On the Relationship with Unlinkability [10]). The unlinkability property, as defined in [10], requires that a signature obtained by first pre-signing and then adapting is indistinguishable from a signature obtained by directly signing the message. Unlinkability implies witness hiding, as the simulator Sim can be replaced by the signing algorithm. However, witness hiding does not imply unlinkability. To see this, consider a witness hiding adaptor signature scheme AS with a simulator Sim. If we modify the pre-sign algorithm so that it signs $(m||0)$ instead of m, and adjust the verification and adaptation algorithms accordingly, then the scheme still satisfies witness hiding. However, the unlinkability property does not hold in this modified scheme, since the pre-sign-then-adapt mode returns a signature for $(m||0)$, while the direct sign mode returns a signature for m.

Though unlinkability [10] is strictly stronger than the witness hiding property defined in this work, it is still reasonable to introduce the *weaker* definition of witness hiding.

- In some applications, we may want to make a pre-sign-and-adapt signature be *distinguishable* from a directly signed signature (while hiding the witness from just the adapted signature at the same time). For example, Alice may use the same signing secret key in both a blockchain transaction system (where Alice pre-signs transactions only) and a daily authentication system (where Alice signs messages directly only), and she wants to distinguish all signatures in the blockchain systems so that she can trace and analyze her behaviors. The unlinkability property is overengineered in this scenario.
- It is essential to explore the minimal requirement for practical AS where the witness is hidden from just the adapted signatures. As shown in this work, such *weaker* definition allows us to design witness hiding AS for all NP relations.

– The unlinkability defined in [10] is very strong, since the adversary is able to select (Y, y) by itself. To achieve unlinkability, [10] requires the strong random-self reducibility of the underlying NP relations, and therefore the instantiations are limited to number theoretical problems (DL, LWE, etc.). It is very challenging to design unlinked AS schemes for arbitrary relations, where the pre-sign-and-adapt signature (in which the instance is specified by the adversary, i.e., $\mathsf{PreSignAdapt}(sk, m, Y, y)$) is indistinguishable from a normal one (in which there is no instance as input at all, i.e., $\mathsf{Sign}(sk, m)$). This means that the instance Y can be re-randomized or eliminated during the adaption, which seems inherently requires some special structure of the underlying NP relation. It is unknown whether unlinked AS schemes for any NP relations exist.

4 Generic Construction of Adaptor Signatures from Signatures and Trapdoor Commitments with a Specific Adaptable Message

Let $\mathcal{M}$ be a message space, and there is a fixed message $m_{\mathbf{0}} \in \mathcal{M}$ (e.g., the all-zero bit string). Let $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme with the message space $\mathcal{M}$. Let R be an NP relation, and any $(Y, y) \in \mathcal{R}$ forms a trapdoor commitment scheme $\mathsf{TC} = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Ver}, \mathsf{TdOpen})$ with a specific adaptable message $m_{\mathbf{0}}$, where Y is the commitment key and y is the trapdoor, and $\mathsf{TC.Gen}(1^\lambda)$ just returns $(Y, y) \leftarrow \mathsf{Sample}(R)$.

Our adaptor signature scheme AS with the message space $\mathcal{M} \setminus \{m_{\mathbf{0}}\}$ is shown in Fig. 7.

Correctness. The correctness of AS consists of three aspects.

– Signature correctness & Pre-signature correctness. These are guaranteed by the correctness of SIG and the correctness of TC (the first property of TC).
– Extraction correctness. This is guaranteed by the trapdoor extractability of TC.

Theorem 2. *If* SIG *has UF-CMA security,* TC *has hiding and trapdoor extractability and* R *is a hard relation, then the adaptor signature scheme* AS *in Fig. 7 has UF-CMA security, witness extractability, pre-signature adaptability and witness hiding.*

Proof. **UF-CMA security**. Let $(m^*, \sigma^* = (\bar{\sigma}^*, Y^*, c^*, d^*))$ be $\mathcal{A}$'s final forgery in the unforgeability experiment (cf. Def. 12), and $m^* \in \mathcal{M} \setminus \{m_{\mathbf{0}}\}$. Recall that for $\mathcal{A}$ to win, it must hold that

$\underline{\mathsf{Gen}(1^\lambda)}$: $(pk, sk) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$ Return (pk, sk) $\underline{\mathsf{Sign}(sk, m)}$: $(Y, y) \leftarrow \mathsf{Sample}(R)$ $(c, d) \leftarrow \mathsf{TC.Com}(Y, m)$ $\bar{\sigma} \leftarrow \mathsf{SIG.Sign}(sk, (m, Y, c))$ Return $\sigma := (\bar{\sigma}, Y, c, d)$ $\underline{\mathsf{Ver}(pk, m, \sigma)}$: Parse $\sigma = (\bar{\sigma}, Y, c, d)$ If $\mathsf{TC.Ver}(Y, c, m, d) = 0$: return 0 Return $\mathsf{SIG.Ver}(pk, (m, Y, c), \bar{\sigma})$	$\underline{\mathsf{pSign}(sk, m, Y)}$: $(c, d_{\mathbf{0}}) \leftarrow \mathsf{TC.Com}(Y, m_{\mathbf{0}})$ $\bar{\sigma} \leftarrow \mathsf{SIG.Sign}(sk, (m, Y, c))$ Return $\tilde{\sigma} := (\bar{\sigma}, Y, c, d_{\mathbf{0}})$ $\underline{\mathsf{pVer}(pk, m, Y, \tilde{\sigma})}$: Parse $\tilde{\sigma} = (\bar{\sigma}, Y, c, d_{\mathbf{0}})$ If $\mathsf{TC.Ver}(Y, c, m_{\mathbf{0}}, d_{\mathbf{0}}) = 0$: return 0 Return $\mathsf{SIG.Ver}(pk, (m, Y, c), \bar{\sigma})$ $\underline{\mathsf{Adapt}(pk, m, \tilde{\sigma}, y)}$: Parse $\tilde{\sigma} = (\bar{\sigma}, Y, c, d_{\mathbf{0}})$ $d \leftarrow \mathsf{TC.TdOpen}(y, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m)$ Return $\sigma := (\bar{\sigma}, Y, c, d)$ $\underline{\mathsf{Ext}(pk, m, Y, \tilde{\sigma}, \sigma)}$: Parse $\tilde{\sigma} = (\bar{\sigma}', Y', c', d_{\mathbf{0}})$ and $\sigma = (\bar{\sigma}, Y, c, d)$ If $(\bar{\sigma}' \neq \bar{\sigma}) \vee (Y' \neq Y) \vee (c' \neq c)$: return $\perp$ Return $\mathsf{TC.Ext}(Y, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m, d)$

Fig. 7. Generic construction of adaptor signatures from signatures and trapdoor commitments with a specific adaptable message $m_{\mathbf{0}}$

1. $\mathsf{SIG.Ver}(pk, (m^*, Y^*, c^*), \bar{\sigma}^*) = 1$ and $\mathsf{TC.Ver}(Y^*, c^*, m^*, d^*) = 1$, and
2. $\mathcal{A}$ never queries $\textsc{Sign}(m^*)$, and
3. (a) either $\mathcal{T}[m^*] = \varnothing$ (i.e., $\mathcal{A}$ never asks $\textsc{pSign}(m^*, Y)$ for any Y), or
   (b) for all $(Y, \tilde{\sigma} = (\bar{\sigma}, Y, c, d)) \in \mathcal{T}[m^*]$, it holds that $Y \in \mathcal{Y}$ (i.e., $\mathcal{A}$ only queries $\textsc{pSign}(m^*, Y)$ for Y whose witness is unknown to it).

We first analyze the case $1 \wedge 2 \wedge (a)$. It is easy to see that in this case, the challenger $\mathcal{C}$ does not sign a message of the form $(m^*, \cdot, \cdot)$ when answering the pre-signing oracle $\textsc{pSign}$ and the signing oracle $\textsc{Sign}$. Therefore, we can easily construct a reduction algorithm to break the UF-CMA security of the underlying SIG.

Then we analyze the case $1 \wedge 2 \wedge (b)$. We divide it into the following two subcases.

(i) For all $(Y, \tilde{\sigma} = (\bar{\sigma}, Y, c, d)) \in \mathcal{T}[m^*]$, $(Y, c) \neq (Y^*, c^*)$.
Similarly, this means that $\mathcal{C}$ does not sign a message in the form of (m^*, Y^*, c^*) when answering the pre-signing oracle $\textsc{pSign}$. Therefore, $\mathcal{A}$ breaks the UF-CMA security of the underlying SIG.

(ii) There exists $(Y, \tilde{\sigma} = (\bar{\sigma}, Y, c, d_{\mathbf{0}})) \in \mathcal{T}[m^*]$ such that $(Y, c) = (Y^*, c^*)$. Recall that the message space defined in AS is $\mathcal{M} \setminus \{m_{\mathbf{0}}\}$. Therefore $m^* \neq m_{\mathbf{0}}$. Besides, we have $\mathsf{TC.Ver}(Y, c, m_{\mathbf{0}}, d_{\mathbf{0}}) = 1$ due to the correctness of AS. Combined with the fact that $\mathsf{TC.Ver}(Y^*, c^*, m^*, d^*) = 1$

and $(Y, c) = (Y^*, c^*)$, we can extract a witness y' via $\mathsf{TC.Ext}(Y^*, c^*, m_{\mathbf{0}}, d_{\mathbf{0}}, m^*, d^*)$. Since $Y^* \in \mathcal{Y}$ (i.e., the corresponding witness of Y^* is unknown to $\mathcal{A}$), $\mathcal{A}$ breaks the one-wayness of the hard relation R.

The UF-CMA security holds as a result.

Witness extractability. Let (m^*, σ^*) be $\mathcal{A}$'s final output in the witness extractability experiment (cf. Def. 13), and $\sigma^* = (\bar{\sigma}^*, Y^*, c^*, d^*)$. Recall that for $\mathcal{A}$ to win, we have

1. $\mathsf{SIG.Ver}(pk, (m^*, Y^*, c^*), \bar{\sigma}^*) = 1$ and $\mathsf{TC.Ver}(Y^*, c^*, m^*, d^*) = 1$, and $\mathcal{A}$ never asks $\textsc{Sign}(m^*)$.
2. For all $(Y, \tilde{\sigma} = (\bar{\sigma}, Y, c, d_{\mathbf{0}})) \in \mathcal{T}[m^*]$, $(Y^*, \mathsf{TC.Ext}(Y, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m^*, d^*)) \notin R$ (i.e., the witness extraction fails for all $(Y, \tilde{\sigma}) \in \mathcal{T}[m^*]$).

Since $\mathcal{A}$ never queries $\textsc{Sign}(m^*)$, and $\bar{\sigma}^*$ is a valid signature w.r.t. (m^*, Y^*, c^*), there must exist an item $(Y, \tilde{\sigma} = (\bar{\sigma}, Y, c, d_{\mathbf{0}})) \in \mathcal{T}[m^*]$ s.t., $Y = Y^*$ and $c = c^*$, as otherwise $\mathcal{A}$ would break the UF-CMA security of the underlying SIG scheme. Then, for that $(Y, \tilde{\sigma}) \in \mathcal{T}[m^*]$, we have $\mathsf{SIG.Ver}(pk, (m^*, Y, c), \bar{\sigma}) = 1$ and $\mathsf{TC.Ver}(Y, c, m_{\mathbf{0}}, d_{\mathbf{0}}) = 1$ due to the correctness of AS. Therefore, from the fact that $\mathsf{TC.Ver}(Y^*, c^*, m^*, d^*) = 1$, $\mathsf{TC.Ver}(Y = Y^*, c = c^*, m_{\mathbf{0}}, d_{\mathbf{0}}) = 1$ and $m^* \neq m_{\mathbf{0}}$, the extraction algorithm $\mathsf{TC.Ext}(Y^*, c^*, m_{\mathbf{0}}, d_{\mathbf{0}}, m^*, d^*)$ will always return a witness y satisfying $(Y^*, y) \in R$, which completes the proof of witness extractability.

Pre-signature adaptability. Let $\tilde{\sigma} = (\bar{\sigma}, Y, c, d_{\mathbf{0}})$ be a pre-signature such that $\mathsf{pVer}(pk, m, Y, \tilde{\sigma}) = 1$. Namely, $\mathsf{SIG.Ver}(pk, (m, Y, c), \bar{\sigma}) = 1$ and $\mathsf{TC.Ver}(Y, c, m_{\mathbf{0}}, d_{\mathbf{0}}) = 1$. Assume $(Y, y) \in R$, the adaption algorithm will return $\sigma = (\bar{\sigma}, Y, c, d)$ where $d \leftarrow \mathsf{TC.TdOpen}(y, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m)$. Obviously $\mathsf{SIG.Ver}(pk, (m, Y, c), \bar{\sigma}) = 1$ still holds. Furthermore, due to the correctness of TC, we have $\mathsf{TC.Ver}(Y, c, m, d) = 1$, and the pre-signature adaptability holds consequently.

Witness hiding. To prove the witness hiding property, we have to design a simulator Sim such that it can simulate an adapted signature given (m, Y), which is indistinguishable from a signature generated by the pre-sign-and-adaption paradigm with the knowledge of y. (Recall that the witness hiding property assumes that the secret key sk for signing is also given to Sim as an input.)

We design Sim similarly to the signing algorithm Sign, with the only difference being that Sim uses a fixed Y instead of sampling a new Y.

Then we argue that a simulated signature $\sigma_1 = (\bar{\sigma}_1, Y, c_1, d_1)$ is indistinguishable from a pre-sign-and-adapt signature $\sigma_0 = (\bar{\sigma}_0, Y, c_0, d_0)$. Notice that $\bar{\sigma}_1 \leftarrow \mathsf{SIG.Sign}(sk, (m, Y, c_1))$ and $\bar{\sigma}_0 \leftarrow \mathsf{SIG.Sign}(sk, (m, Y, c_0))$. Therefore, it is sufficient to prove that (c_1, d_1) is indistinguishable from (c_0, d_0).

- In the simulated signature, (c_1, d_1) is computed via $(c_1, d_1) \leftarrow \mathsf{TC.Com}(Y, m)$.
- In the pre-sign-and-adaption signature, (c_0, d_0) is computed via $(c_0, \boldsymbol{d_0}) \leftarrow \mathsf{Com}(Y, \boldsymbol{m_0})$ and $d_0 \leftarrow \mathsf{TC.TdOpen}(y, c_0, \boldsymbol{m_0}, \boldsymbol{d_0}, m)$.

According to the adaption indistinguishability of TC, (c_1, d_1) and (c_0, d_0) are indistinguishable, and the witness hiding of AS holds consequently.

□

5 Trapdoor Commitments for Any NP Relation

In this section we show a trapdoor commitment with a specific adaptable message for the Hamiltonian cycle problem, a well-known NP complete problem. Combined with Theorem 2, we obtain adaptor signatures for any NP relation.

Zero-knowledge proof for the Hamiltonian cycle problem. Let us first recall the zero-knowledge protocol for the Hamiltonian cycle problems by Blum [7, 14]. Let G be a graph, and $H \subseteq G$ be a Hamiltonian cycle, i.e., a witness of Hamiltonian graph instance G. The zero-knowledge protocol between the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ is shown as follows.

1. $\mathcal{P}$ randomly samples a permutation π and gets $G' := \pi(G)$. Then $\mathcal{P}$ commits the adjacency matrix of G' and sends the commitments $com_{G'}$ to $\mathcal{V}$.
2. After receiving $com_{G'}$, $\mathcal{V}$ sends a random bit $b \xleftarrow{\$} \{0, 1\}$.
3. $\mathcal{P}$ responses as follows.
   - If $b = 0$, then $\mathcal{P}$ sends all openings of $com_{G'}$, and the permutation π to $\mathcal{V}$.
   - If $b = 1$, then $\mathcal{P}$ sends all openings of $com_{H'}$ to $\mathcal{V}$, where $H' := \pi(H)$ is a Hamiltonian cycle of G'.
4. $\mathcal{V}$ checks as follows.
   - If $b = 0$, then $\mathcal{V}$ checks $com_{G'}$ are commitments of the adjacency matrix of $\pi(G)$.
   - If $b = 1$, then $\mathcal{V}$ checks $com_{G'}$ include commitments of the adjacency matrix of H', and H' is a Hamiltonian cycle.

Theorem 3 ([7, 14]). *If the commitment scheme has statistical biding and (computational) hiding, then the above protocol is a zero-knowledge proof protocol with soundness error 1/2.*

Now we show the zero-knowledge proof (Sigma) protocol can be transferred into a bit trapdoor commitment with special adaptable message $m_{\mathbf{0}} = 0$. At a high-level, the transform follows the proof of the equivalence of Sigma protocols and chameleon hashes by Bellare and Ristov [4]. Specifically, the commitment, challenge, and response in a sigma protocol correspond to the commitment, message, and opening in a trapdoor commitment scheme, respectively, with the witness of the Sigma protocol serving as the trapdoor in the trapdoor commitment scheme. In more detail, the bit trapdoor commitment scheme is as follows. Here, $\mathcal{M} = \{0, 1\}$ and $m_{\mathbf{0}} = 0$.

- $\mathsf{Gen}(1^\lambda)$. Randomly sample a Hamiltonian graph G with a Hamiltonian cycle $H \subseteq G$ as its witness. Return $(ck, td) := (G, H)$.
- $\mathsf{Com}(ck, m \in \{0, 1\})$.
  - If $m = 0$, then randomly sample a permutation π, and commit the adjacency matrix of $G' := \pi(G)$ to get $com_{G'}$ and the corresponding openings $d_{G'}$. Return $(c, d) := (com_{G'}, (\pi, d_{G'}))$.
  - If $m = 1$, then randomly generate a Hamiltonian graph G' with a Hamiltonian cycle H', and commit the adjacency matrix of G' to get $com_{G'}$ and the corresponding openings $d_{G'}$. Return $(c, d) := (com_{G'}, (H', d_{H'}))$.
- $\mathsf{Ver}(ck, c, m, d)$.
  - If $m = 0$, parse $(c, d) = (com_{G'}, (\pi, d_{G'}))$. Return 1 if $com_{G'}$ are the commitments of $\pi(G)$, and 0 otherwise.
  - If $m = 1$, parse $(c, d) = (com_{G'}, (H', d_{H'}))$. Return 1 if $com_{G'}$ includes commitments of a Hamiltonian cycle H', and 0 otherwise.
- $\mathsf{TdOpen}(td = H, c, m_{\mathbf{0}} = 0, d_{\mathbf{0}}, m)$.
  - If $m = 0$, return $d_{\mathbf{0}}$ directly.
  - If $m = 1$, parse $(c, d_{\mathbf{0}}) := (com_{G'}, (\pi, d_{G'}))$. Return $\perp$ if $com_{G'}$ are not the commitments of $\pi(G)$. Otherwise, let $H' := \pi(H)$, and $d_{H'} \subseteq d_{G'}$ be the openings w.r.t. the adjacency matrix of H'. Return $d := (H', d_{H'})$.

The correctness is implied by the completeness of the zero-knowledge protocol for the Hamiltonian cycle problem.

Theorem 4. *If the underlying commitment scheme has statistical biding and computational hiding, then the above constructed trapdoor commitment scheme with specific adaptable message $m_{\mathbf{0}} = 0$ has (computational) hiding, trapdoor extractability, and adaption indistinguishability.*

Proof. **Hiding**. This follows directly from the hiding property of the underlying commitment scheme. Namely, for any two graphs G_0 and G_1 of the same size (i.e., with the same number of vertices and edges), the commitments com_{G_0} and com_{G_1} are computationally indistinguishable.

Trapdoor Extractability. Let $(c = com_{G'}, m_{\mathbf{0}} = 0, d_{\mathbf{0}} = (\pi, d_{G'}))$ and $(c = com_{G'}, m = 1, d = (H', d_{H'}))$ be two commitment-message-opening tuples. On one hand, $\mathsf{Ver}(ck, c, m_{\mathbf{0}}, d_{\mathbf{0}}) = 1$ implies that $com_{G'}$ is a commitment of graph $\pi(G)$. On the other hand, $\mathsf{Ver}(ck, c, m, d) = 1$ implies that $com_{G'}$ contains a commitment of Hamiltonian cycle H'. Since the commitment scheme is statistically biding, $H' \subseteq \pi(G)$ holds with overwhelming probability. Therefore, $\pi^{-1}(H') \subseteq G$ is a Hamiltonian cycle, which finishes the proof of trapdoor extractability.

Adaption Indistinguishability. The adaption indistinguishability property requires that the distribution (c, d), where $(c, d_{\mathbf{0}}) \leftarrow \mathsf{Com}(ck, m_{\mathbf{0}})$ and $d \leftarrow \mathsf{TdOpen}(td, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m = 1)$, is indistinguishable from the distribution (c, d), where $(c, d) \leftarrow \mathsf{Com}(ck, m = 1)$.

Recall that in both cases (c, d) is in the form of $(com_{G'}, (H', d_{H'}))$.

- In the first case, $G' = \pi(G)$ and $H' = \pi(H)$, where π is a random permutation.
- In the second case, G' is a randomly sampled graph with a Hamiltonian cycle H'.

Since π is a random permutation, $(H', d_{H'})$ distributed identically in both cases. The only difference lies in the parts of $com_{G' \setminus H'}$, which are computationally indistinguishable due to the hiding property of the underlying commitment scheme. □

Extension for Large Message Space . Via the standard hybrid arguement, it is easy to extend the message space from $\{0, 1\}$ to $\{0, 1\}^\ell$ for any polynomial ℓ, and the specific adaptable message $m_{\mathbf{0}}$ is 0^ℓ.

Further Discussion . Though the above-mentioned construction works for any NP relation R, it involves a heavy Karp reduction [17] from R to the

Hamiltonian cycle problem. For commonly used relations in cryptography such as the DL relation, the RSA relation, the LWE relation, and the SIS relation, more efficient trapdoor commitments (with a specific adaptable message) can be constructed from the Schnorr identification scheme [24, 23], the RSA-based Sigma protocol/chameleon hashes [1], the LWE-based Sigma protocols [9, 5] and the SIS-based Sigma protocols [20, 6], respectively. Furthermore, in Appendix A, we show a direct construction of trapdoor commitments for any NP relation with random self-reducibility.

6 Conclusion

In this work we introduce witness hiding for adaptor signatures (AS), which requires that the witness y can be extracted from both a pre-signature and an adapted signature, but not from either of them individually. We propose a generic construction of witness-hiding AS from signatures and trapdoor commitments with a specific adaptable message, a weaker version of trapdoor commitments. Based on the Hamiltonian cycle problem, we also propose a trapdoor commitment scheme with a specific adaptable message, where the commitment key is the Hamiltonian problem instance and the trapdoor is the Hamiltonian cycle witness. Therefore, we prove that the existence of one-way functions implies the existence of witness hiding adaptor signatures for any NP relation.

Further Work. One potential approach to circumvent the heavy Karp reduction [17] is to leverage the multi-party computation in the head (MPCitH) paradigm [16]. Namely, let Π be a secure MPC protocol for $f(\cdot)$, where $f(Y, y_1, ..., y_n)$ is a function with public input Y and private inputs y_i from users U_i for $i \in [n]$, and $f(Y, y_1, ..., y_n)$ outputs 1 if and only if $(Y, \bigoplus_{i\in[n]} y_i) \in R$. Taking the MPCitH paradigm, we immediately obtain a Sigma (zero-knowledge) protocol with special soundness, which is closely related to the witness extractability of AS as discussed earlier. However, converting such a Sigma protocol into a trapdoor commitment scheme with a special adaptable message is not straightforward. In the Sigma protocol, a witness is necessary for generating the commitment (the first message in the protocol) to ensure that there exists a valid response for every possible challenge. On the other hand, in a trapdoor commitment scheme, there is no trapdoor (witness) when generating a commitment for a message. In fact, to construct AS, we require a Sigma protocol of which the commitment does not rely on the witness, which

cannot be satisfied by the MPCitH paradigm. We leave constructing more efficient AS for NP relations from MPCitH as a further work.

Acknowledgements

We thank all anonymous reviewers for their valuable comments. This work was done while the authors were at Purdue University. Xiangyu Liu and Ioannis Tzannetos were funded by AnalytiXIN and Sunday Group, Inc. Vassilis Zikas was funded in part by NSF grant No. 2055599, AnalytiXIN, and Sunday Group, Inc.

References

[1] Ateniese, G., de Medeiros, B.: Identity-based chameleon hash and applications. In: Juels, A. (ed.) FC 2004, Key West, FL, USA, February 9-12, 2004. Revised Papers. Lecture Notes in Computer Science, vol. 3110, pp. 164–180. Springer (2004). https://doi.org/10.1007/978-3-540-27809-2_19, https://doi.org/10.1007/978-3-540-27809-2_19

[2] Aumayr, L., Ersoy, O., Erwig, A., Faust, S., Hostáková, K., Maffei, M., Moreno-Sanchez, P., Riahi, S.: Generalized bitcoin-compatible channels. IACR Cryptol. ePrint Arch. p. 476 (2020), https://eprint.iacr.org/2020/476

[3] Aumayr, L., Ersoy, O., Erwig, A., Faust, S., Hostáková, K., Maffei, M., Moreno-Sanchez, P., Riahi, S.: Generalized channels from limited blockchain scripts and adaptor signatures. In: ASIACRYPT 2021. vol. 13091, pp. 635–664. Springer (2021). https://doi.org/10.1007/978-3-030-92075-3_22, https://doi.org/10.1007/978-3-030-92075-3_22

[4] Bellare, M., Ristov, T.: A characterization of chameleon hash functions and new, efficient designs. J. Cryptol. **27**(4), 799–823 (2014). https://doi.org/10.1007/S00145-013-9155-8, https://doi.org/10.1007/s00145-013-9155-8

[5] Benhamouda, F., Krenn, S., Lyubashevsky, V., Pietrzak, K.: Efficient zero-knowledge proofs for commitments from learning with errors over rings. In: Pernul, G., Ryan, P.Y.A., Weippl, E.R. (eds.) ESORICS 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9326, pp. 305–325. Springer (2015). https://doi.org/10.1007/978-3-319-24174-6_16, https://doi.org/10.1007/978-3-319-24174-6_16

[6] Beullens, W.: Sigma protocols for mq, PKP and sis, and fishy signature schemes. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12107, pp. 183–211. Springer (2020). https://doi.org/10.1007/978-3-030-45727-3_7, https://doi.org/10.1007/978-3-030-45727-3_7

[7] Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians. vol. 1, p. 2. Citeseer (1986)

[8] Bursuc, S., Mauw, S.: Contingent payments from two-party signing and verification for abelian groups. In: CSF 2022, Haifa, Israel, August 7-10, 2022. pp. 195–210. IEEE (2022). https://doi.org/10.1109/CSF54842.2022.9919674, https://doi.org/10.1109/CSF54842.2022.9919674

[9] Corrigan-Gibbs, H.: Lattice-based signatures. Presentation Slides (3 2024), https://65610.csail.mit.edu/2024/lec/l14-latsig.pdf, accessed on May 2024

[10] Dai, W., Okamoto, T., Yamamoto, G.: Stronger security and generic constructions for adaptor signatures. In: Isobe, T., Sarkar, S. (eds.) INDOCRYPT 2022, Kolkata, India, December 11-14, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13774, pp. 52–77. Springer (2022). https://doi.org/10.1007/978-3-031-22912-1_3, https://doi.org/10.1007/978-3-031-22912-1_3

[11] Erwig, A., Faust, S., Hostáková, K., Maitra, M., Riahi, S.: Two-party adaptor signatures from identification schemes. In: Garay, J.A. (ed.) PKC 2021, Virtual Event, May 10-13, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12710, pp. 451–480. Springer (2021). https://doi.org/10.1007/978-3-030-75245-3_17, https://doi.org/10.1007/978-3-030-75245-3_17

[12] Erwig, A., Riahi, S.: Deterministic wallets for adaptor signatures. In: Atluri, V., Pietro, R.D., Jensen, C.D., Meng, W. (eds.) ESORICS 2022, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13555, pp. 487–506. Springer (2022). https://doi.org/10.1007/978-3-031-17146-8_24, https://doi.org/10.1007/978-3-031-17146-8_24

[13] Esgin, M.F., Ersoy, O., Erkin, Z.: Post-quantum adaptor signatures and payment channel networks. In: Chen, L., Li, N., Liang, K., Schneider, S.A. (eds.) ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12309, pp. 378–397. Springer (2020). https://doi.org/10.1007/978-3-030-59013-0_19, https://doi.org/10.1007/978-3-030-59013-0_19

[14] Goldreich, O.: The Foundations of Cryptography - Volume 1: Basic Techniques. Cambridge University Press (2001). https://doi.org/10.1017/CBO9780511546891, http://www.wisdom.weizmann.ac.il/%7Eoded/foc-vol1.html

[15] Goldreich, O.: The Foundations of Cryptography - Volume 2: Basic Applications. Cambridge University Press (2004). https://doi.org/10.1017/CBO9780511721656, http://www.wisdom.weizmann.ac.il/%7Eoded/foc-vol2.html

[16] Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge proofs from secure multiparty computation. SIAM J. Comput. **39**(3), 1121–1152 (2009). https://doi.org/10.1137/080725398, https://doi.org/10.1137/080725398

[17] Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA. pp. 85–103. The IBM Research Symposia Series, Plenum Press, New York (1972). https://doi.org/10.1007/978-1-4684-2001-2_9, https://doi.org/10.1007/978-1-4684-2001-2_9

[18] Klamti, J.B., Hasan, M.A.: Post-quantum two-party adaptor signature based on coding theory. Cryptogr. **6**(1), 6 (2022). https://doi.org/10.3390/CRYPTOGRAPHY6010006, https://doi.org/10.3390/cryptography6010006

[19] Krawczyk, H., Rabin, T.: Chameleon hashing and signatures. IACR Cryptol. ePrint Arch. p. 10 (1998), http://eprint.iacr.org/1998/010

[20] Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Cramer, R. (ed.) PKC 2008, Barcelona, Spain, March 9-12, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4939, pp. 162–179. Springer (2008). https://doi.org/10.1007/978-3-540-78440-1_10, https://doi.org/10.1007/978-3-540-78440-1_10

[21] Poelstra, A.: Scriptless scripts. Presentation Slides (3 2017), https://download.wpsoftware.net/bitcoin/wizardry/mw-slides/2017-03-mit-bitcoin-expo/slides.pdf, accessed on May 2024
[22] Poelstra, A.: Mimblewimble and scriptless scripts. Presentation Slides (1 2018), https://download.wpsoftware.net/bitcoin/wizardry/mw-slides/2018-01-10-rwc/slides.pdf, accessed on May 2024
[23] Schnorr, C.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO '89, Santa Barbara, California, USA, August 20-24, 1989, Proceedings. Lecture Notes in Computer Science, vol. 435, pp. 239–252. Springer (1989). https://doi.org/10.1007/0-387-34805-0_22, https://doi.org/10.1007/0-387-34805-0_22
[24] Schnorr, C.: Efficient identification and signatures for smart cards (abstract). In: Quisquater, J., Vandewalle, J. (eds.) EUROCRYPT '89, Houthalen, Belgium, April 10-13, 1989, Proceedings. Lecture Notes in Computer Science, vol. 434, pp. 688–689. Springer (1989). https://doi.org/10.1007/3-540-46885-4_68, https://doi.org/10.1007/3-540-46885-4_68
[25] Tairi, E., Moreno-Sanchez, P., Maffei, M.: Post-quantum adaptor signature for privacy-preserving off-chain payments. In: Borisov, N., Díaz, C. (eds.) FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part II. Lecture Notes in Computer Science, vol. 12675, pp. 131–150. Springer (2021). https://doi.org/10.1007/978-3-662-64331-0_7, https://doi.org/10.1007/978-3-662-64331-0_7
[26] Tairi, E., Moreno-Sanchez, P., Schneidewind, C.: Ledgerlocks: A security framework for blockchain protocols based on adaptor signatures. In: Meng, W., Jensen, C.D., Cremers, C., Kirda, E. (eds.) CCS 2023, Copenhagen, Denmark, November 26-30, 2023. pp. 859–873. ACM (2023). https://doi.org/10.1145/3576915.3623149, https://doi.org/10.1145/3576915.3623149
[27] Tu, B., Zhang, M., Yu, C.: Efficient ecdsa-based adaptor signature for batched atomic swaps. In: Susilo, W., Chen, X., Guo, F., Zhang, Y., Intan, R. (eds.) ISC 2022, Bali, Indonesia, December 18-22, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13640, pp. 175–193. Springer (2022). https://doi.org/10.1007/978-3-031-22390-7_12, https://doi.org/10.1007/978-3-031-22390-7_12

A Trapdoor Commitments from Random Self-Reducible Relations

Let R be a hard relation and $\mathcal{L}_R$ be the language defined by R. We recall the definition of random self-reducibility in [10].

Definition 16. *A relation R is random self-reducible, if there are three additional algorithms* $\mathsf{ReRandIns}$, $\mathsf{ReRandWit}$, $\mathsf{Recover}$ *and a randomness space Ω, such that for any $(Y, y) \in R$, the following properties hold.*

1. *If r distributes uniformly over Ω, then (Y', y') distributes identically to a sample by* $\mathsf{Sample}(R)$*, where* $Y' \leftarrow \mathsf{ReRandIns}(Y, r)$*,* $y' \leftarrow \mathsf{ReRandWit}(y, r)$*.*
2. *For any $r \in \Omega$, it holds that $(Y', y') \in R$, where* $Y' \leftarrow \mathsf{ReRandIns}(Y, r)$*,* $y' \leftarrow \mathsf{ReRandWit}(y, r)$*.*
3. *For any $r \in \Omega$,* $Y' \leftarrow \mathsf{ReRandIns}(Y, r)$ *and $(Y', y') \in R$, it holds that $(Y, y'') \in R$, where* $y'' \leftarrow \mathsf{Recover}(y', r)$*.*

Now we show the construction of trapdoor commitment with specific adaptable message $m_{\mathbf{0}} = 0$.

- $(ck, td) \leftarrow \mathsf{Gen}(1^\lambda)$. Sample $(Y, y) \leftarrow \mathsf{Sample}(R)$ and return $(ck, td) := (Y, y)$.
- $(c, d) \leftarrow \mathsf{Com}(ck, m \in \{0, 1\})$.
  - If $m = 0$, then sample $r \xleftarrow{\$} \Omega$, compute $Y' \leftarrow \mathsf{ReRandIns}(Y, r)$, and return $(c, d) = (Y', r)$.
  - If $m = 1$, then $(Y', y') \leftarrow \mathsf{Sample}(R)$, and return $(c, d) = (Y', y')$.
- $0/1 \leftarrow \mathsf{Ver}(ck, c, m, d)$.
  - If $m = 0$, let $c = Y'$ and $d = r$. Return 1 if $Y' = \mathsf{ReRandIns}(Y, r)$.
  - If $m = 1$, let $c = Y'$ and $d = y'$. Return 1 if $(Y', y') \in R$.
- $d \leftarrow \mathsf{TdOpen}(td, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m)$.
  - If $m = 0$, return $d_{\mathbf{0}}$.
  - If $m = 1$, let $d_{\mathbf{0}} = r$. Return $y' \leftarrow \mathsf{ReRandWit}(y, r)$.

Correctness. The correctness is straightforward due to Property 2.

Theorem 5. *If the relation R has random self-reducibility, then the above constructed trapdoor commitment scheme with a specific adaptable message $m_{\mathbf{0}} = 0$ has hiding, trapdoor extractability, and adaption indistinguishability.*

Proof. **Hiding**. This is implied by Property 1, i.e., for random r and any $(Y, y) \in R$, $Y' \leftarrow \mathsf{ReRandIns}(Y, r)$ distributes identically to the instance sampled via $(Y, y) \leftarrow \mathsf{Sample}(R)$.

Trapdoor Extractability. This is implied by Property 2, and the extraction algorithm is simply $\mathsf{Recover}$. Specifically, given Y', y', r such that $(Y', y') \in R$ and $Y' = \mathsf{ReRandIns}(Y, r)$, $y'' \leftarrow \mathsf{Recover}(y', r)$ is a witness for Y.

Adaption Indistinguishability. The adaption indistinguishability requires that the distribution (c, d) where $(c, d_{\mathbf{0}}) \leftarrow \mathsf{Com}(ck, m_{\mathbf{0}})$ and $d \leftarrow \mathsf{TdOpen}(td, c, m_{\mathbf{0}}, d_{\mathbf{0}}, m = 1)$, is indistinguishable from the distribution (c, d) where $(c, d) \leftarrow \mathsf{Com}(ck, m = 1)$. This is directly implied by Property 1 of R. □