IEEE Open Journal of the Computer Society

Received 7 August 2023; accepted 27 August 2023. Date of publication 29 August 2023; date of current version 13 September 2023. The review of this paper was arranged by Associate Editor Zibin Zheng.

Digital Object Identifier 10.1109/OJCS.2023.3309836

An Identity-Based Adaptor Signature Scheme and its Applications in the Blockchain System

ZIJIAN BAO [1], DEBIAO HE [2,3] (Member, IEEE), CONG PENG [4], MIN LUO [5,6], AND KIM-KWANG RAYMOND CHOO [7] (Senior Member, IEEE)

[1]School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[2]Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China
[3]School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[4]School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[5]School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[6]Shanghai Key Laboratory of Privacy-Preserving Computation, Matrix Elements Technologies, Shanghai 200120, China
[7]Department of Information Systems and Cyber Security, Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249 USA

CORRESPONDING AUTHORS: DEBIAO HE; MIN LUO (e-mails: hedebiao@163.com; mluo@whu.edu.cn).

This work was supported in part by the Shandong Provincial Key Research and Development Program under Grant 2021CXGC010107, in part by the National Natural Science Foundation of China under Grants 62172307, U21A20466, and 62272350, in part by the New 20 Project of Higher Education of Jinan under Grant 202228017, in part by the Special Project on Science and Technology Program of Hubei Provience under Grants 2020AEA013 and 2021BAA025, and in part by the Fundamental Research Funds for the Central Universities under Grant 2042023KF0203. The work of Kim-Kwang Raymond Choo was supported by Cloud Technology Endowed Professorship.

ABSTRACT Adaptor signature, as a new emerging cryptographic primitive, has become one promising method to mitigate the *scalability* issue on blockchain. It can transform an incomplete signature into a complete signature by revealing the witness of a pre-set hard relation, which can be applied to atomic swap, payment channel, payment hub, and other blockchain scenarios. Recently, a general transformation for constructing adaptor signatures has been proposed for some signature schemes with specific structures, e.g., Schnorr, ECDSA, SM2 signatures. However, we note that there is no identity-based adaptor signature method so far. In this article, we put forward an adaptor signature scheme for the identity-based signature scheme in the IEEE P1363 standard. Then, we formally prove the security of our scheme under the random oracle model. We also present the computation and communication costs, compared with other adaptor signatures. Finally, we show our scheme's potential use in atomic swaps and payment channel networks of blockchain.

INDEX TERMS Adaptor signature, IEEE P1363, identity-based signature, payment channel.

I. INTRODUCTION

The emergence of blockchains [1] makes it promising to deploy enormous decentralized applications, greatly improving their security and reliability. It put forward a decentralized payment model, consisting of participating computing devices (nodes) to reach a consensus. However, its growth faces one crucial element: *scalability*. In fact, most existing blockchain systems suffer from the *poor* transaction throughput [2]. For example, Bitcoin suffers from its poor performance around 10 transactions per second [3], which seriously hinders the popularization of blockchain applications.

A payment channel [4], [5], [6], [7], [8] is a solution for conducting efficient and low-cost transactions on a blockchain. Payment channels address this by creating a private channel between participants, allowing them to conduct multiple transactions off-chain and then submit the final state to the blockchain when needed, reducing transaction costs and confirmation times. First, the two users initiate a channel by securing a specific amount of coins on the blockchain within a jointly controlled account. Following that, they engage in a series of off-chain transactions, exchanging authenticated messages to facilitate the process. Finally, when concluding the channel, they publicly announce the outcomes of their transactions on the ledger. However, such an approach relies on a Turing-complete scripting language, which allows for the implementation of complex and programmable smart contracts on the blockchain.

One of the key techniques is the hash-locking scripts [9]. The traditional Hash Locking mechanism is a way to implement conditional payments in payment channels, where participants must share a preimage hash beforehand and provide the corresponding unlocking data in each transaction to verify the payment conditions. However, this method has limitations, such as the need to reveal the hash preimage in each transaction, which can increase the transaction volume and on-chain burden of the channel.

Adapter Signature (AS) [10] is an alternative technique that replaces the traditional Hash Locking mechanism in payment channels to improve efficiency and flexibility. It is one of promising solutions to build a Layer 2 protocol[1] for extending the performance of original blockchains, without changing the blockchain itself. It was first proposed by Poelstra [10] and formalized by Aumayr et al. [7]. Specifically, it can output a pre-signature under a hard relation, and the pre-signature can be converted into a complete signature by a *publisher* having the witness of a hard relation, and the transformed signature can be verified by the traditional process.

Intuitively, the adaptor signatures should have two properties: i) only users who know the witness can transform a pre-signature into a complete signature; ii) any user can extract the witness by leveraging the pre-signature and the complete signature. Based on the above two properties, the adaptor signatures provide huge potential for addressing *scalability* issues in blockchain and have been proved to be used in practice. For example, payment channels [11], [12], atomic swaps [13], [14], or payment channel networks (PCNs) [5], [15].

A. MOTIVATION

Identity-based cryptography has become an important component of the modern public-key cryptosystem. Instead of using public keys, the users can directly make use of others' user-friendly identity (e.g., a user's e-mail address). It is commonly utilized in the Internet of Things (IoT), E-mail and other fields to reduce the overhead of managing keys. Recently, the academia and industry have observed that the identity-based system can also be useful in the blockchain, especially in the blockchain-based IoT scenario. Using identity instead of public keys, the user management can be easily conducted and malicious behavior can be viably tracked. Wan et al. [16] proposed HIBEChain, a hierarchical identity-based blockchain system for large-scale IoT. ChainMaker [17], which is a famous blockchain team in China, also claimed that their platform supports hierarchical identity-based encryption. Even a certificateless consortium blockchain [18] has been proposed in recent years. However, we remark that no concrete identity-based adaptor signature scheme has been proposed in theory.

The identity-based adaptor signature scheme offers several compelling advantages over traditional signature schemes. Its user-friendly approach allows users to utilize easily recognizable information, such as email addresses or usernames, as their public keys, simplifying the key management process and enhancing user experience. Additionally, the scheme ensures efficient key distribution without the need for a complex public key infrastructure (PKI), making it particularly suitable for applications with a large number of users or devices. Its scalability enables seamless handling of a growing user base, making it an ideal solution for scenarios with widespread adoption. Further, in practical applications such as identity-based blockchain, it can play an important role in atomic swaps and payment channel networks by moving the on-chain operations to off-chain. Thus, we ask the question "*can we design an identity-based adaptor signature scheme to make up for the lack of theory?*".

B. OUR CONTRIBUTIONS

In our work, 1) we present an identity-based adaptor signature scheme. We start from the signature scheme in the IEEE P1363 standard [19] to construct our adaptor signatures, and give the specific construction of it. 2) We formally prove our scheme's security by sequences games. 3) We present the computation and communication costs, compared with other adaptor signatures. 4) We show our proposed adaptor signature scheme can be used in atomic swaps and payment channel networks.

C. ORGANIZATION

In Section II, we review recent literature on adaptor signatures. In Section III, we give the preliminaries, such as digital signature, IEEE P1363 standard for identity-based signature, hard relation and non-interactive zero-knowledge (NIZK) proof. In Section IV, we provide the basic concepts. Then, in Section V, we present the proposed adaptor signature. In Sections VI and VII, we analyze its security and offer the experimental results. In Section VIII, we show two potential applications of our scheme: atomic swap and payment channel networks. Finally, we conclude this article in Section IX.

II. RELATED WORK

In 2017, Polestra [10] proposed the concept of scriptless scripts, which was formally defined as adaptor signature. Malavolta et al. [15] presented scriptless constructions for schnorr/ECDSA signature schemes, and designed a provably secure anonymous multi-hop lock mechanism based on them, so as to realize a secure and privacy-protected payment channel network. However, this work does not define the adaptor signature as an independent primitive, and does not provide an independent formal proof. Fournier [20] tried to formalize the adaptor signature as an instantiation of a one-time verifiable encrypted signature. However, in their definition of the *unforgeability* game, in the challenge phase, the adversary lacks the capacity to obtain pre-signatures, which is unsuitable for practical applications.

Aumayr et al. [7] solved the above problem and gave the formal definition of adaptor signature, constructed adaptor

[1] Blockchain Layer 2 refers to the auxiliary framework or protocol built on the existing blockchain system.

signature schemes focused on Schnorr and ECDSA signature schemes with a formal proof, named as Schnorr-AS and ECDSA-AS, and designed a payment channel instance, which can provide richful off-chain channel operations. Moreno-Sanchez et al. [6] proposed an adaptor algorithm based on linkable ring signature, which provides expressiveness and interoperability for Monero, and alleviates the problem of low efficiency caused by large-scale applications. Tairi et al. [8] designed an anonymous atomic lock mechanism using adaptor signature to build a payment channel hub scheme, meeting the security and privacy requirements in the application process. Erwig et al. [21] further put forward two-party adaptor signatures from identification schemes. It can be used in escrow protocols for blockchain, which needs a collaboration between the two parties. Thyagarajan et al. [22] introduced a lockable signature scheme similar to the adaptor signature, and built a payment channel network suitable for any signature scheme. Lockable signature can only be pre-signed with a given witness. BLS signatures can benefit from its structure, while most other signatures rely on secure multi-party computation (MPC) protocols. Peng et al. [23] proposed an adaptor signature scheme SM2-AS based on the Chinese Commercial Cryptographic Standard SM2 signature algorithm. The security of the scheme was proven under the random oracle model, demonstrating its fulfillment of properties.

To resist post quantum attacks, Esgin et al. [24] proposed the first post-quantum adaptor signature algorithm, LAS. They used the standard lattice assumptions SIS and LWE. However, this kind of structure has a large communication overhead. To solve the above problems, Tairi et al. [25] proposed a post-quantum adaptor signature algorithm IAS based on isogenies, and gave security proof for quantum adversaries. Compared with LAS, IAS can reduce storage space by three times, but it needs more computation time.

III. PRELIMINARIES

We present the notions and cryptographic primitives which will be used later.

A. NOTIONS

The security parameter is denoted by 1^n, and probabilistic polynomial time is abbreviated as $\mathcal{PPT}$. We also abbreviate key generation center as KGC. Also, $x \leftarrow_R \mathbb{Z}_p$ denotes x is randomly selected from $\mathbb{Z}_p$ and $\epsilon(n)$ indicates the negligible function.

Definition 1 (Bilinear Map [26]): Given three groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$, a bilinear map satisfies that $\tilde{e}$: $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. It satisfies the following properties:

- *Bilinear:* Input $a, b \in Z_q^*$ and any $X \in \mathbb{G}_1$, any $Y \in \mathbb{G}_2$, $\tilde{e}(a \cdot X, b \cdot Y) = \tilde{e}(X, Y)^{ab}$.
- *Non-degenerate:* We have $X \in \mathbb{G}_1$, $Y \in \mathbb{G}_2$ satisfies $\tilde{e}(X, Y) \neq 1_{\mathbb{G}_T}$.
- *Efficient computability:* For any $X \in \mathbb{G}_1$, any $Y \in \mathbb{G}_2$, $\tilde{e}(X, Y)$ is computable efficiently.

B. DIGITAL SIGNATURE

A digital signature scheme [27] $\Pi_{Sig} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ is defined as follows.

- `Gen`(1^n)*:* It inputs 1^n for the system security parameter, outputs (pk, sk).
- `Sign`(m, sk)*:* It inputs a message $m \in \{0, 1\}^*$ and sk, outputs a signature σ.
- `Verify`(m, σ, pk)*:* It inputs m, σ and pk, outputs a bit $b \in \{0, 1\}$ for `invalid/valid`.

A digital signature scheme is *correct*, if for any m and any (pk, sk), we have $\mathsf{Verify}(\mathsf{Sign}(m, sk), m, pk) = 1$. Then, we give the definition of *existential unforgeability under chosen message attack* (`EUF-CMA`) and *strong existential unforgeability under chosen message attack* (`SUF-CMA`).

Definition 2 (`EUF-CMA`)*:* For any $\mathcal{PPT}$ adversary $\mathcal{A}$, and a digital signature scheme. We define a experiment $\mathtt{SigForge}_{\mathcal{A},\Pi_{Sig}}$ as follows:

- $(pk, sk) \leftarrow Gen(1^n)$*:* The challenger uses the $Gen()$ function to generate (pk, sk).
- $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot)}(pk)$*:* $\mathcal{A}$ first accesses `Sign` oracle $\mathcal{O}_S(m_i)$ to obtain the corresponding signature σ_i, sets $\mathcal{Q} := \mathcal{Q} \cup \{m_i\}$, where $\mathcal{Q}$ is firstly initialized to an empty list. Then, $\mathcal{A}$ outputs a forged pair (m^*, σ^*).
- *Outputs* $\{0, 1\}$*:* Once the forged signature pair (m^*, σ^*) satisfies the conditions $m^* \notin \mathcal{Q} \wedge Verify(m^*, \sigma^*) = 1$, output 1; otherwise 0.

For any $\mathcal{PPT}$ $\mathcal{A}$, there is a negligible function ϵ for any n that satisfies $\Pr[\mathrm{SigForge}_{\mathcal{A},\Pi_{Sig}} = 1] \leq \epsilon(n)$, then the signature is secure under `EUF-CMA`.

Definition 3 (`SUF-CMA`)*:* For any $\mathcal{PPT}$ adversary $\mathcal{A}$, and a digital signature scheme. We define a experiment $\mathtt{strongSigForge}_{\mathcal{A},\Pi_{Sig}}$ as follows:

- $(pk, sk) \leftarrow Gen(1^n)$*:* The challenger uses the $Gen()$ function to generate (pk, sk).
- $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot)}(pk)$*:* $\mathcal{A}$ first accesses `Sign` oracle $\mathcal{O}_S(m_i)$ to obtain the corresponding signature σ_i, sets $\mathcal{Q} := \mathcal{Q} \cup \{(m_i, \sigma_i)\}$, where $\mathcal{Q}$ is firstly initialized to an empty list. Then, $\mathcal{A}$ outputs a forged pair (m^*, σ^*).
- *Outputs* $\{0, 1\}$*:* Once the forged signature pair (m^*, σ^*) satisfies the conditions $(m^*, \sigma^*) \notin \mathcal{Q} \wedge Verify(m^*, \sigma^*) = 1$, output 1; otherwise 0.

For any $\mathcal{PPT}$ $\mathcal{A}$, there is a negligible function ϵ for any n that satisfies $\Pr[\mathrm{strongSigForge}_{\mathcal{A},\Pi_{Sig}} = 1] \leq \epsilon(n)$, then the signature is secure under `SUF-CMA`.

C. HARD RELATION

Given a binary relation $R \subseteq S \times W$ for a statement/witness pair $(Y, y) \subseteq S \times W$, let $L_R := \{Y \in S \mid \exists y \in W, s.t. (Y, y) \in R\}$ be the language for describing this relationship. If L_R satisfies the following conditions, then we say that L_R is a hard relation [21].

- There is a $\mathcal{PPT}$ algorithm $\mathrm{Gen}(1^n)$, taking security parameter 1^n, outputting a pair $(Y, y) \in R$.
- For the relation R, it is decidable in polytime.

- Given a statement $Y \in L_R$, it is negligible for any $\mathcal{PPT}$ adversary $\mathcal{A}$ to generate a valid witness y satisfying $(Y, y) \in R$.

D. ADAPTOR SIGNATURE

According to [7], an adaptor signature consist of the following algorithms with a signature scheme with $\Sigma = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$.

1) $\tilde{\sigma} \leftarrow pSign(m, (pk, sk), Y)$: Given $m \in \{0, 1\}^*$, a key pair (pk, sk), $Y \in L_R$ as inputs, output a pre-signature $\tilde{\sigma}$.
2) $b \leftarrow pVerify(m, \tilde{\sigma}, pk, Y)$: Given a message $m \in \{0, 1\}^*$, $\tilde{\sigma}$, a public key pk and a statement $Y \in L_R$ as inputs, output $b \in \{0, 1\}$ for `invalid/valid`.
3) $\sigma \leftarrow Adapt(m, \tilde{\sigma}, pk, y)$: Given a message $m \in \{0, 1\}^*$, a pre-signature $\tilde{\sigma}$, a public key pk, a witness y as inputs, output a signature σ.
4) $y \leftarrow Ext(\sigma, \tilde{\sigma}, Y)$: Given $\sigma, \tilde{\sigma}$ and Y as inputs, output y satisfying $(Y, y) \in R$.

E. IEEE P1363 STANDARD FOR IDENTITY-BASED SIGNATURE

We discuss the `IEEE P1363 standard for identity-based signature` scheme [19]. We use 2 hash functions $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ as well as $H_2 : \{0, 1\}^* \times \mathbb{G}_T \rightarrow \mathbb{Z}_q^*$. The steps are as follows:

1) *Setup:* Given 1^n as input, output a public parameter set $\mathbb{PP}$:
   a) Generate the cyclic groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$, set the bilinear map: $\tilde{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$, where q is the same order for both $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$.
   b) Select 2 random generators $Q_1 \in \mathbb{G}_1$ and $Q_2 \in \mathbb{G}_2$.
   c) Choose $s \leftarrow_R \mathbb{Z}_q^*$, set s as the KGC's master secret key, then compute $P = s \cdot Q_2$, $g = \tilde{e}(Q_1, Q_2)$.
   d) Output the public parameter set $\mathbb{PP} = \{P, \tilde{e}, g, Q_1, Q_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$.
2) *Extract:* Given a users's ID, $\mathbb{PP}$, and the KGC's secret key s as inputs, output a user's d_{ID}:
   a) Calculate the user's identity $h_{ID} = H_1(ID)$.
   b) Output the user's private key $d_{ID} = (s + h_{ID})^{-1} \cdot Q_1$.
3) *Sign:* Given m, $\mathbb{PP}$, and d_{ID} as inputs, output a signature σ:
   a) Choose $r \leftarrow_R \mathbb{Z}_q^*$, then calculate $R = g^r$.
   b) Calculate $h = H_2(m, R)$ and $S = (r + h) \cdot d_{ID}$.
   c) Output $\sigma = (h, S)$.
4) *Verify:* Given a message m, a signature σ, the user's identity ID as inputs, output $b \in \{0, 1\}$ for `invalid/valid`.
   a) Compute $h_{ID} = H_1(ID)$.
   b) Compute $R' = \tilde{e}(S, h_{ID} \cdot Q_2 + P) \cdot g^{-h}$.
   c) If $h = H_2(m, R')$, output 1; otherwise 0.

F. NON-INTERACTIVE ZERO-KNOWLEDGE PROOF

For an NP language L_R, a NIZK proof system [28], [29] with *online extractor* contains two $\mathcal{PPT}$ algorithms (`Prove`, `Verify`). Note that we need the *online extractor*, rather than the classical *soundness*. It is because we need the simulator to extract the witness in the later security proof.

- *Prove*(Y, y)*:* take the statement Y, the witness y as inputs, output a proof π.
- *Verify*(Y, π)*:* take the statement Y, proof π as inputs, output a bit $b \in \{0, 1\}$ for `invalid/valid`.

A NIZK proof needs to have the following three properties:

- *Completeness:* For each $(Y, y) \in R$, $\Pr[\pi \leftarrow \mathsf{Prove}(Y, y), \mathsf{Verify}(Y, \pi) = 1] \geq 1 - \epsilon(n)$
- *Online extractor:* For any $\mathcal{PPT}$ adversary $\mathcal{A}$,

$$\Pr\begin{bmatrix} \pi \leftarrow \mathsf{Prove}(Y, y) \\ \tilde{y} \leftarrow \mathcal{A}(Y, \pi) : \\ (Y, \tilde{y}) \notin L_R \end{bmatrix} \leq \epsilon(n)$$

- *Zero-knowledge:* For a $\mathcal{PPT}$ simulator $\mathcal{S}$, it satisfies that:

$$\left| \begin{array}{l} \Pr\begin{bmatrix} (Y, y) \leftarrow \mathcal{A}_1(\cdot) & (Y, y) \in R \wedge \\ \pi \leftarrow \mathsf{Prove}(Y, y) & : \mathcal{A}_2(\pi) = 1 \end{bmatrix} \\ -\Pr\begin{bmatrix} (Y, td) \leftarrow \mathcal{A}_1(\cdot) & (Y, y) \in R \wedge \\ \tilde{\pi} \leftarrow \mathcal{S}(Y, td) & : \mathcal{A}_2(\tilde{\pi}) = 1 \end{bmatrix} \end{array} \right| \leq \epsilon(n)$$

IV. BASIC CONCEPTS

We further propose the formal definition and security model.

A. FORMAL DEFINITION

In short, adaptor signature is a *two-step* signature algorithm: the signer uses the private key sk pre-signs a message m with a statement Y to obtain a pre-signature, then the person with the witness y can convert a pre-signature into a complete form.

Definition 4 (Identity-Based Adaptor Signature): Given a hard relation R and a digital signature scheme $\Pi_{Sig} =$ (Gen, Sign, Verify), an adaptor signature $\Xi_{R,\Pi_{Sig}}$ consists of 8 algorithms (`Setup, GenR, Extract, pSign, pVerify, Adapt, Verify, Ext`). The details are described as follows:

- $\mathbb{PP} \leftarrow Setup(1^n)$*:* Given 1^n as an input, output a public parameter set $\mathbb{PP}$.
- $(Y, y, \pi) \leftarrow GenR(1^n)$*:* Given 1^n as an input, output a statement/witness pair $(Y, y) \in R$ and a zero-knowledge proof π, where R is hard relation.
- $d_{ID} \leftarrow Extract(ID, s)$*:* Given a user's ID and the KGC's secret key as inputs, output a user's d_{ID}.
- $\tilde{\sigma} \leftarrow pSign(m, d_{ID}, Y, \pi)$*:* Given $m \in \{0, 1\}^*$, the user's d_{ID}, $Y \in L_R$ and the zero-knowledge proof π as inputs, output a pre-signature $\tilde{\sigma}$.
- $b \leftarrow pVerify(m, \tilde{\sigma}, ID, Y)$*:* Given a message $m \in \{0, 1\}^*$, $\tilde{\sigma}$, the user's ID and $Y \in L_R$ as inputs, output $b \in \{0, 1\}$ for `invalid/valid`.
- $\sigma \leftarrow Adapt(\tilde{\sigma}, y)$*:* Given a $\tilde{\sigma}$, y as inputs, output a signature σ.
- $b \leftarrow Verify(m, \sigma, ID)$*:* Given $m \in \{0, 1\}^*$, σ and ID as inputs, output $b \in \{0, 1\}$ for `invalid/valid`.

- $y \leftarrow Ext(\sigma, \tilde{\sigma}, Y)$: Given σ, $\tilde{\sigma}$ and Y as inputs, output y satisfying $(Y, y) \in R$.

B. SECURITY MODEL

We present the conrrectness and security model of our proposed scheme.

Definition 5 (Pre-signature Correctness): For any m, any $(Y, y) \in R$.

$$\Pr\left[\begin{array}{lcr} \mathbb{PP} \leftarrow \text{Setup}(1^n) & & \\ (Y, y, \pi) \leftarrow \text{GenR}(1^n) & & \text{pVerify}(m, \tilde{\sigma}, ID, Y) = 1 \wedge \\ d_{ID} \leftarrow \text{Extract}(ID, s) & : & \text{Verify}(m, \sigma, ID) = 1 \wedge \\ \tilde{\sigma} := \text{pSign}(m, d_{ID}, Y, \pi) & & (Y, y') \in R \\ \text{y'} := \text{Ext}(\sigma, \tilde{\sigma}, Y) & & \end{array}\right] = 1$$

Then, we discuss the security properties. Similar to EUF-CMA, an adaptor signature additionally needs to satisfy that even given a message m's pre-signature m^*, it is difficult for any $\mathcal{A}$ to generate a forged signature σ of m. By capturing this, we give the following definition, aEUF-CMA security, where "a" represents "adaptor".

*Definition 6 (**aEUF-CMA** Security):* $\Xi_{R,\Pi_{Sig}}$ is aEUF-CMA secure if for any $\mathcal{PPT}$ $\mathcal{A}$, the probability of winning the aSigForge$_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}$ experiment is negligible, namely $\Pr[\text{aSigForge}_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}(1^n) = 1] \leq \epsilon(n)$. Following is a definition of the experiment:

1) *Initializes* $\mathcal{Q}$*:* The challenger creates an empty message query list $\mathcal{Q}$.
2) $(Y, y, \pi) \leftarrow$ GenR*:* The challenger creates a statement/witness pair $(Y, y) \in R$ and a zero-knowledge proof π, where R is hard relation.
3) $d_{ID} \leftarrow Extract(ID, s)$*:* The challenger uses the Extract() to create the user's d_{ID}.
4) $m^* \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot),\mathcal{O}_{pS}(\cdot)}(ID)$*:* Any adversary $\mathcal{A}$ can access Sign oracle $\mathcal{O}_S(m_i)$ and pSign oracle $\mathcal{O}_{pSign}(m_j, Y, \pi)$, and obtains the corresponding signatures σ_i and $\tilde{\sigma}_j$, where $\mathcal{Q} := \mathcal{Q} \cup \{m_i\}$, $\mathcal{Q} := \mathcal{Q} \cup \{m_j\}$. Then, $\mathcal{A}$ outputs m^*, where $m^* \notin \mathcal{Q}$.
5) $\tilde{\sigma} \leftarrow \text{pSign}_{d_{ID}}(m^*, Y, \pi)$*:* The challenger pre-signs the message m, outputs $\tilde{\sigma}$.
6) $\sigma^* \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot),\mathcal{O}_{pS}(\cdot)}(\tilde{\sigma}, Y)$*:* $\mathcal{A}$ also accesses Sign oracle $\mathcal{O}_S(m_i)$ and pSign oracle $\mathcal{O}_{pS}(m_j, Y, \pi)$, and obtains the corresponding signatures σ_i and $\tilde{\sigma}_j$, where $\mathcal{Q} := \mathcal{Q} \cup \{m_i\}$, $\mathcal{Q} := \mathcal{Q} \cup \{m_j\}$. Given $\tilde{\sigma}$ and Y, $\mathcal{A}$ outputs a forged signature σ^*.
7) *Outputs* $\{0, 1\}$*:* If the forged pair (m^*, σ^*) satisfies the conditions $m^* \notin \mathcal{Q} \wedge \text{Verify}(m^*, \sigma^*, ID) = 1$, the experiment outputs 1; otherwise 0.

Different from *Pre-signature conrrectness*, we further require that even a malicious produced pre-signature can be converted into a complete form, which is defined next.

Definition 7 (Pre-signature Adaptability): $\Xi_{R,\Pi_{Sig}}$ satisfies pre-signature adaptability if we have $\Pr[\text{Verfiy}(m, \text{Adapt}(\tilde{\sigma}, y))] = 1$ for any $\tilde{\sigma}$ where $\text{pVerify}(m, \tilde{\sigma}, ID, Y) = 1$.

We also give the definition of witness extractability, which ensures that there is a $\mathcal{PPT}$ algorithm to extract the witness y given a pair $(\sigma, \tilde{\sigma})$ for (m, Y).

Definition 8 (Witness Extractability): $\Xi_{R,\Pi_{Sig}}$ satisfies witness extractability if for any $\mathcal{PPT}$ $\mathcal{A}$, the probability of winning the aWinExt$_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}$ experiment is negligible, $\Pr[\text{aWinExt}_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}(1^n) = 1] \leq \epsilon(n)$. Following is a definition of the experiment:

1) *Initializes* $\mathcal{Q}$*:* The challenger creates an empty message query list $\mathcal{Q}$.
2) $d_{ID} \leftarrow Extract(ID, s)$*:* The challenger uses the Extract() to create the user's d_{ID}.
3) $(m^*, Y, \pi) \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot),\mathcal{O}_{pS}(\cdot)}(ID)$*:* Any adversary $\mathcal{A}$ can access Sign oracle $\mathcal{O}_{sign}(m_i)$ and pSign oracle $\mathcal{O}_{pS}(m_j, Y, \pi)$, and obtains the corresponding signatures σ_i and $\tilde{\sigma}_j$, where $\mathcal{Q} := \mathcal{Q} \cup \{m_i\}$, $\mathcal{Q} := \mathcal{Q} \cup \{m_j\}$. Then, $\mathcal{A}$ outputs m and Y, where $m^* \notin \mathcal{Q}$.
4) $\tilde{\sigma} \leftarrow pSign_{d_{ID}}(m^*, Y, \pi)$*:* The challenger pre-signs the message m, outputs $\tilde{\sigma}$.
5) $\sigma^* \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot),\mathcal{O}_{pS}(\cdot)}(\tilde{\sigma})$*:* $\mathcal{A}$ also accesses Sign oracle $\mathcal{O}_S(m_i)$ and pSign oracle $\mathcal{O}_{pS}(m_j, Y, \pi)$, and obtains the corresponding signatures σ_i and $\tilde{\sigma}_j$, where $\mathcal{Q} := \mathcal{Q} \cup \{m_i\}$, $\mathcal{Q} := \mathcal{Q} \cup \{m_j\}$. Given $\tilde{\sigma}$, $\mathcal{A}$ outputs σ^*.
6) $y' := Ext(\sigma^*, \tilde{\sigma}, Y)$*:* Given σ^*, $\tilde{\sigma}$ and Y, the challenger extracts y'.
7) *Outputs* $\{0, 1\}$*:* If the pair (m^*, σ^*, Y, y') satisfies the conditions $m^* \notin \mathcal{Q} \wedge \text{Verify}(m^*, \sigma^*) = 1 \wedge (Y, y') \notin R$, the experiment outputs 1; otherwise 0.

V. PROPOSED ADAPTOR SIGNATURE SCHEME

We provide the proposed identity-based adaptor signature scheme $\Xi_{R,\Pi^{P1363}_{Sig}}$ for the IEEE P1363 standard. It contains a set of algorithms {Setup, GenR, Extract, pSign, pVerify, Adapt, Verify, Ext} (also see Fig. 1).

1) $\mathbb{PP} \leftarrow Setup(1^n)$*:*
   a) Generate the cyclic groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$, set the bilinear map: $\tilde{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$, where q is the same order for both $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$.
   b) Select two random generators $Q_1 \in \mathbb{G}_1$ and $Q_2 \in \mathbb{G}_2$.
   c) Choose $s \leftarrow_R \mathbb{Z}_q^*$, set s as the KGC's master secret key, then compute $P = s \cdot Q_2$, $g = \tilde{e}(Q_1, Q_2)$.
   d) Output the public parameter set $\mathbb{PP} = \{P, \tilde{e}, g, Q_1, Q_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$.
2) $(Y, y, \pi) \leftarrow GenR(1^n)$*:*
   a) Choose $y \in \mathbb{G}_1$ as the witness, calculate $Y = \tilde{e}(y, h_{ID} \cdot Q_2 + P)$, where $h_{ID} = H_1(ID)$ is the user's identity.
   b) Set the hard relation as $R_Y := \{(Y, y) : Y = \tilde{e}(y, h_{ID} \cdot Q_2 + P)\}$.
   c) Generate a zero-knowledge proof $\pi = \text{Prove}(Y, y)\{Y = \tilde{e}(y, h_{ID} \cdot Q_2 + P)\}$.
   d) Output the statement/witness pairs (Y, y) and the zero-knowledge proof π.

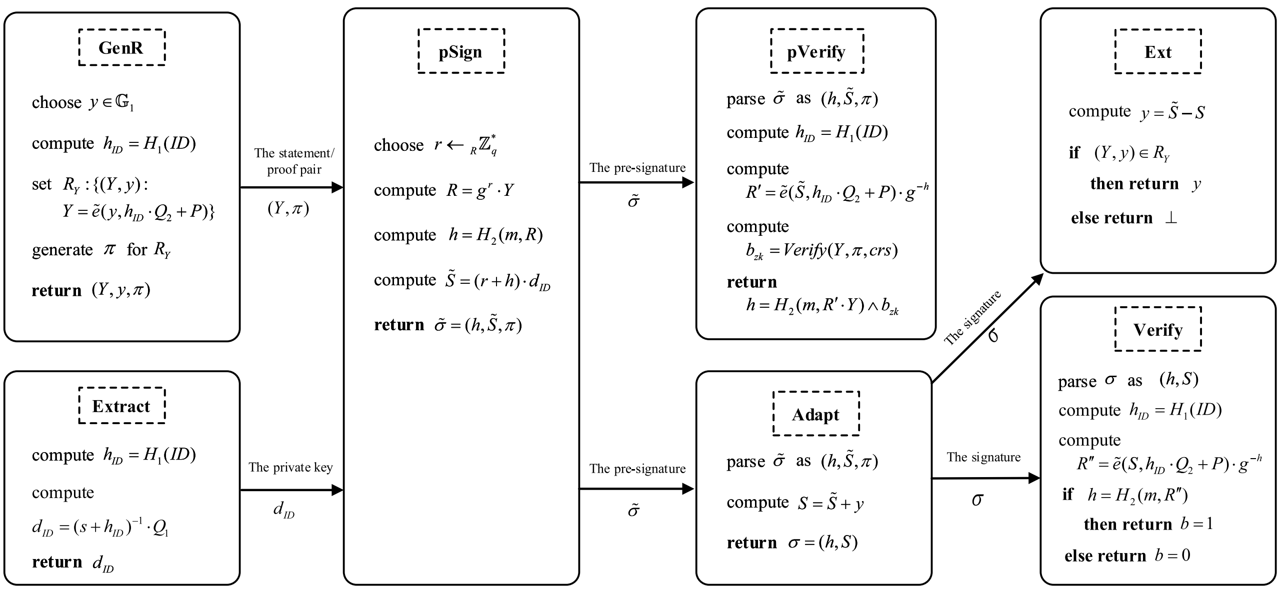


FIGURE 1. Our proposed scheme. (As the `Setup` algorithm outputs the system parameter set for other algorithms, we do not show it here).

3) $d_{ID} \leftarrow Extract(ID, s)$:
a) Calculate the user's identity $h_{ID} = H_1(ID)$.
b) Output the user's private key $d_{ID} = (s + h_{ID})^{-1} \cdot Q_1$.
4) $\tilde{\sigma} \leftarrow pSign(m, d_{ID}, Y, \pi)$:
a) Choose $r \leftarrow_R \mathbb{Z}_q^*$, then calculate $R = g^r \cdot Y$.
b) Calculate $h = H_2(m, R)$ and $\tilde{S} = (r + h) \cdot d_{ID}$.
c) Output the pre-signature $\tilde{\sigma} = (h, \tilde{S}, \pi)$.
5) $b \leftarrow pVerify(m, \tilde{\sigma}, ID, Y)$:
a) Parse $\tilde{\sigma}$ as $(h, \tilde{S}, \pi)$.
b) Compute $h_{ID} = H_1(ID)$.
c) Compute $R' = \tilde{e}(\tilde{S}, h_{ID} \cdot Q_2 + P) \cdot g^{-h}$.
d) Compute $b_{zk} = \mathsf{Verify}(Y, \pi)$.
e) If $(h = H_2(m, R' \cdot Y) \wedge b_{zk} = 1)$, then output $b = 1$; otherwise $b = 0$.
6) $\sigma \leftarrow Adapt(\tilde{\sigma}, y)$:
a) Parse $\tilde{\sigma}$ as $(h, \tilde{S}, \pi)$.
b) Compute $S = \tilde{S} + y$.
c) Output the signature $\sigma = (h, S)$.
7) $b \leftarrow Verify(m, \sigma, ID)$:
a) Parse σ as (h, S).
b) Compute $h_{ID} = H_1(ID)$.
c) Compute $R'' = \tilde{e}(S, h_{ID} \cdot Q_2 + P) \cdot g^{-h}$.
c) If $h = H_2(m, R'')$, then output $b = 1$; otherwise $b = 0$.
8) $y \leftarrow Ext(\sigma, \tilde{\sigma}, Y)$:
a) Compute $y = S - \tilde{S}$.
b) Check whether $(Y, y) \in R_Y$.
c) If yes, return y; otherwise, return $\perp$.

A. INSTANTIATION OF ZERO KNOWLEDGE PROOF

We want to prove $\pi = \mathsf{Prove}(Y, y)\{Y = \tilde{e}(y, h_{ID} \cdot Q_2 + P)\}$. We choose a random number $t \leftarrow_R \mathbb{Z}_q^*$, let $y' = y + t \cdot Q_1$, $C = h_{ID} \cdot Q_2 + P$. Then, we know

$$\tilde{e}(y', C) = Y \cdot \tilde{e}(Q_1, C)^t.$$

Let $T = \tilde{e}(y', C)/Y$, $F = \tilde{e}(Q_1, C)$. Next, we can transfer the original one π into $\pi_{new} = \mathsf{Prove}((T, F), t)\{T = F^t\}$. The specific process is as follows:

1) $Prove(Y, y)$:
a) The prover chooses a challenge $c \leftarrow_R \mathbb{Z}_q^*$.
b) The prover computes $T_1 = F^c$.
c) The prover computes $e = H(T_1)$.
d) The prover computes $z = c - e \cdot t$ and sends $\pi := (e, z)$ to the verifier.
2) $Verify(Y, \pi)$:
a) The verifier checks if $e = H(F^z \cdot T^e)$.

Equivalence of π and π_{new}. We demonstrate the equivalence between the original and converted forms of π. The converted form is $\pi_{new} = \mathsf{Prove}((T, F), t)\{T = F^t\}$. We have the following equation holds.

$$\begin{aligned} & T = F^t \\ \Longleftrightarrow\ & \tilde{e}(y', C)/Y = \tilde{e}(Q_1, C)^t \\ \Longleftrightarrow\ & \tilde{e}(y', C) = Y \cdot \tilde{e}(Q_1, C)^t \\ \Longleftrightarrow\ & \tilde{e}(y + t \cdot Q_1, C) = Y \cdot \tilde{e}(Q_1, C)^t \\ \Longleftrightarrow\ & \tilde{e}(y, C) = Y \\ \Longleftrightarrow\ & \tilde{e}(y, h_{ID} \cdot Q_2 + P) = Y \end{aligned}$$

So far, we have shown that they are equivalent. Since the converted SOK_1 now takes the form of a standard discrete logarithmic proof, its correctness can be verified, and we shall not delve into the details further.

VI. SECURITY PROOF

In this section, we give the security proof of each property.

Lemma 1 (Pre-signature Adaptability): $\Xi_{R,\Pi_{Sig}^{P1363}}$ satisfies pre-signature adaptability.

For any m, any $y \in \mathbb{G}_1, r \in \mathbb{Z}_q^*, S \in \mathbb{G}_1$, We know $Y = \tilde{e}(y, h_{ID} \cdot Q_2 + P)$ and $S = \tilde{S} + y$. Assuming the $\texttt{pVerify}(m, \tilde{\sigma}, ID, Y) = 1$, we can obtain that:

$$\begin{aligned} h &= H_2(m, R' \cdot Y) \\ &= H_2(m, \tilde{e}(\tilde{S}, h_{ID} \cdot Q_2 + P) \cdot g^{-h} \cdot Y) \\ &= H_2(m, \tilde{e}(\tilde{S} + y, h_{ID} \cdot Q_2 + P) \cdot g^{-h}) \\ &= H_2(m, \tilde{e}(S, h_{ID} \cdot Q_2 + P) \cdot g^{-h}) \\ &= H_2(m, R'') \end{aligned}$$

Obviously, the signature value $\sigma = (h, S)$ is a valid signature for message m.

Lemma 2 (Pre-signature Correctness): The identity-based adaptor signature scheme for the IEEE P1363 standard $\Xi_{R,\Pi_{Sig}^{P1363}}$ satisfies pre-signature correctness.

Proof: We first fix any ID and $y \in \mathbb{G}_1$, define $d_{ID} = (s + h_{ID})^{-1} \cdot Q_1$, $h_{ID} = H_1(ID)$ and $Y = \tilde{e}(y, h_{ID} \cdot Q_2 + P)$. We execute the $\texttt{Psign}(m, d_{ID}, Y, \pi)$ to obatin $\tilde{\sigma} = (h, \tilde{S})$, where $h = H_2(m, R)$ and $\tilde{S} = (r + h) \cdot d_{ID}$ for some $r \in \mathbb{Z}_q^*$. As,

$$\begin{aligned} h &= H_2(m, R' \cdot Y) \\ &= H_2(m, \tilde{e}(\tilde{S}, h_{ID} \cdot Q_2 + P) \cdot g^{-h} \cdot Y) \\ &= H_2(m, \tilde{e}((r + h) \cdot d_{ID}, h_{ID} \cdot Q_2 + s \cdot Q_2) \cdot g^{-h} \cdot Y) \\ &= H_2(m, \tilde{e}(\frac{(r + h)}{(s + h_{ID})} \cdot Q_1, (s + h_{ID}) \cdot Q_2) \cdot g^{-h} \cdot Y) \\ &= H_2(m, \tilde{e}((r + h) \cdot Q_1, Q_2) \cdot g^{-h} \cdot Y) \\ &= H_2(m, g^{r+h} \cdot g^{-h} \cdot Y) \\ &= H_2(m, g^r \cdot Y) \\ &= H_2(m, R) \end{aligned}$$

Then, by the correctness of NIZK, we have $\texttt{pVerify}(m, \tilde{\sigma}, ID, Y) = 1$. By Lemma 1, we have $\texttt{Verify}(m, \sigma, ID) = 1$ for some $\sigma = (h, S) = (h, \tilde{S} + y) = \textsf{Adapt}(\tilde{\sigma}, y)$. At last,

$$Ext(\sigma, \tilde{\sigma}, Y) = S - \tilde{S} = (\tilde{S} + y) - \tilde{S} = y$$

which completes the proof. □

Lemma 3 (aEUF–CMA Security): If the identity-based signature scheme in the IEEE P1363 standard is SUF−CMA secure and R is a hard relation, then $\Xi_{R,\Pi_{Sig}^{P1363}}$ is aEUF−CMA secure.

Proof: We want to prove $\Xi_{R,\Pi_{Sig}^{P1363}}$ is aEUF−CMA secure by reduction to the strong unforgeability of P1363 identity-based signature scheme. Let $\mathcal{A}$ be the adversary who performs the aSigForge game, we consider a simulator $\mathcal{S}$ who performs the strong unforgeability game for the identity-based signature scheme in the IEEE P1363 standard. $\mathcal{S}$ has access to the $\texttt{Sign}$ oracle Sign^{P1363}, the random oracle $\mathcal{H}^{P1363}$, $\mathcal{S}$ uses these oracles to simulate the oracle queries for $\mathcal{A}$, i.e., the $\texttt{Sign}$, $\texttt{pSign}$, and $\texttt{Random}$ oracle queries.

One challenge we need to solve is how to simulate $\mathcal{O}_{pS}$ queries, as $\mathcal{S}$ only knows the complete signatures from its $\mathcal{O}_S$, thus needs a way to obtain the pre-signature. $\mathcal{S}$ needs to know the witness y from the zero-knowledge proof π_Y. Here, we use the *online extractor* property to extract y. Then, we are able to convert the complete signature into a valid pre-signature. Thus, $\mathcal{S}$ completes the simulation of $\mathcal{O}_{pS}$.

We construct security proofs by sequences games [30]. We first describe five games $\texttt{Game 0-4}$. $\texttt{Game 0}$ is the original $\texttt{aSigForge}$ experiment, we will demonstrate that these games are indistinguishable. At last, we prove a simulator can simulate $\texttt{Game 4}$ and use $\mathcal{A}$ to win in the $\texttt{strongSigForge}$ game.

$\texttt{Game 0}$*:* It is the $\texttt{aSigForge}$ game. The A forges a signature σ^* of a message m^*. We have $\Pr[\textsf{Game}_0 = 1] = \Pr[\text{aSigForge}_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}(1^n) = 1]$.

$\texttt{Game 1}$*:* Similar to $\texttt{Game 0}$, the only difference in $\texttt{Game 1}$ is that the simulator $\mathcal{S}$ will verify $\textsf{Adapt}(\tilde{\sigma}, y) = \sigma^*$ after receiving the forged signature σ^* generated by $\mathcal{A}$. If $\texttt{true}$, the game outputs $\perp$.

Claim: Assume $\textsf{Bad}_1$ is the event that $\texttt{Game 1}$ outputs $\perp$, $\Pr[\textsf{Bad}_1] \leq \epsilon_1(n)$.

Proof: As the only difference is that event $\textsf{Bad}_1$ happens, it keeps that $\Pr[\textsf{Game}_1 = 1] \leq \Pr[\textsf{Game}_0 = 1] + \epsilon_1(n)$.

$\texttt{Game 2}$*:* Compared to $\texttt{Game 1}$, $\texttt{Game 2}$ has an additional $\mathcal{O}_{pS}(m, Y, \pi)$. Namely, we add several steps before $\texttt{pSign}$. $\texttt{Game 2}$ extracts a witness y, then checks whether $(Y, y) \in R_Y$. If it does not hold, then the game outputs $\perp$.

Claim: Assume $\textsf{Bad}_2$ is the event that $\texttt{Game 2}$ outputs $\perp$, $\Pr[\textsf{Bad}_2] \leq \epsilon_2(n)$.

Proof: As the *online extractor* property of NIZK, for the witness y extracted from (Y, π), it keeps that $(Y, y) \in R_Y$.

As the only difference is that event $\textsf{Bad}_2$ happens, it keeps that $\Pr[\textsf{Game}_2 = 1] \leq \Pr[\textsf{Game}_1 = 1] + \epsilon_2(n)$.

$\texttt{Game 3}$*:* It extends the previous game to help a simulator $\mathcal{S}$ simulate the $\texttt{pSign}$ queries. It first invokes the $\texttt{Sign}$ algorithm to get a complete signature. Then, it can convert the complete signature into a pre-signature by y. Hence, it keeps that $\Pr[\textsf{Game}_3 = 1] \leq \Pr[\textsf{Game}_2 = 1] + \epsilon_3(n)$.

$\texttt{Game 4}$*:* When getting $\mathcal{A}$'s challenge message m, this game invokes the $\texttt{Sign}$ algorithm to get a complete signature, then transforms it into a pre-signature by using y. As a result, we have the same indistinguishability argument in the prior game, it keeps that $\Pr[\textsf{Game}_4 = 1] \leq \Pr[\textsf{Game}_3 = 1] + \epsilon_3(n)$.

Then, we need to show that there exists $\mathcal{S}$ which can perfectly simulate $\texttt{Game 4}$ and make use of $\mathcal{A}$'s ability to win $\texttt{strongSigForge}$. The steps are as follows.

1) *Sign queries:* When $\mathcal{A}$ querying $\mathcal{O}_S$, $\mathcal{S}$ sends m to the oracle Sign^{P1363} and sends the result to $\mathcal{A}$.

2) *Random Oracle queries:* When $\mathcal{A}$ querying $\mathcal{H}(x)$, if $H[x] = \perp$, $\mathcal{S}$ queries $\mathcal{H}^{P1363}$, otherwise returns $H[x]$.
3) *Pre-Sign queries:*
a) When $\mathcal{A}$ querying $\mathcal{O}_{pS}$, the simulator $\mathcal{S}$ extracts the witness y using the *online extractor* property, sends m to the Sign^{P1363} oracle and parses σ as (h, S).
b) $\mathcal{S}$ computes a pre-signature $(h, \tilde{S}, \pi)$ by calculating $\tilde{S} := S - y$.
4) *Challenge phase:*
a) When $\mathcal{A}$ outputting the challenge message m^*, $\mathcal{S}$ creates (Y, y, π) by `GenR` algorithm, sends m^* to the Sign^{P1363} oracle and parses σ as (h, S).
b) $\mathcal{S}$ computes a pre-signature $(h, \tilde{S}, \pi)$ by calculating $\tilde{S} := S - y$.
c) When $\mathcal{A}$ outputting a forgery σ^*, $\mathcal{S}$ outputs (m^*, σ^*) as its forged message/signature pair.

The difference between the aforementioned simulation and `Game 4` is syntactical. Rather than using the `Sign` and $\mathcal{H}$ oracles, $\mathcal{S}$ utilizes Sign^{P1363} and $\mathcal{H}^{P1363}$. Finally, we need to prove that the simulator $\mathcal{S}$ can use (m^*, σ^*) to win in `strongSigForge`.

Claim: (m^*, σ^*) is a valid forged message/signature pair in `strongSigForge`.

Proof: Before the challenge phase, $\mathcal{A}$ has not queried the $\mathcal{O}_S$ or $\mathcal{O}_{pS}$ on the message m^*. Thus, Sign^{P1363} is is only queried on m^*. As we discussed in `Game 0`, the probability of a event that the σ output by $\mathcal{A}$ is equal to the σ output by Sign^{P1363} is negligible. Thus, Sign^{P1363} has not output σ^* on a input m^*. Subsequently, (m^*, σ^*) is a valid forged message/signature pair in `strongSigForge`.

Finally, we know that the transformations from `Game 0` to `Game 4` is indistinguishable. We can get $\Pr[\text{aSigForge}_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}(1^n) = 1] = \Pr[\mathsf{Game}_0 = 1] \leq \Pr[\mathsf{Game}_4 = 1] + \epsilon_1(n) + \epsilon_2(n) + 2\epsilon_3(n) \leq \Pr[\mathsf{strongSigForge}_{\mathcal{A},\Pi_{Sig}} = 1] + \epsilon_1(n) + \epsilon_2(n) + 2\epsilon_3(n)$. □

Lemma 4 (Witness Extractability): If the identity-based signature scheme in the IEEE P1363 standard is `SUF-CMA` secure and R is a hard relation, $\Xi_{R,\Pi_{Sig}^{P1363}}$ is witness extractable.

Proof: Similar to the previous proof of `aEUF-CMA` security, we prove $\Xi_{R,\Pi_{Sig}^{P1363}}$ is `witness extractability` secure by reduction to the strong unforgeability of the identity-based signature scheme in the IEEE P1363 standard. Let $\mathcal{A}$ be the adversary who runs the aWitExt game, we consider a simulator $\mathcal{S}$ who runs the strong unforgeability game for the identity-based signature scheme in the IEEE P1363 standard. The simulator $\mathcal{S}$ utilizes the $\mathcal{A}$'s ability to win `strongSigForge`.

One difference beween `aWinExt` and `aEUF-CMA` is as follows: In `aEUF-CMA`, the game use `GenR` to generate Y, While, in `aWinExt`, $\mathcal{A}$ is responsible for generating Y. Thus, the witness y is unknown to $\mathcal{S}$. Fortunately, we can employ the same method as in the proof process of Lemma 3, by leveraging the *online extractor* property to extract y.

`Game 0`*:* It is the `aWinExt` game. Given the pre-signature $\tilde{\sigma}$ and m^* and Y generated by A, the adversary A needs to forge a signature σ^*, where $(Y, y) \notin R_Y$, $y := \mathtt{Ext}(\sigma, \tilde{\sigma}, Y)$. We have $\Pr[\mathsf{Game}_0 = 1] = \Pr[\text{aWinExt}_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}(1^n) = 1]$.

`Game 1`*:* Compared to `Game 0`, `Game 1` has an additional $\mathcal{O}_{pS}(m, Y, \pi)$. Namely, we add several steps before `pSign`. This game extracts a witness y, then checks whether $(Y, y) \in R_Y$. If it does not hold, then the game outputs $\perp$.

Claim: Assume Bad_1 is the event that `Game 1` outputs $\perp$, $\Pr[\mathsf{Bad}_1] \leq \epsilon_1(n)$.

Proof: As the *online extractor* property of NIZK, for the witness y extracted from (Y, π), it keeps that $(Y, y) \in R_Y$.

As the only difference is that event Bad_1 happens, it keeps that $\Pr[\mathsf{Game}_1 = 1] \leq \Pr[\mathsf{Game}_0 = 1] + \epsilon_1(n)$.

`Game 2`*:* It extends the previous game to help a simulator $\mathcal{S}$ simulate the `pSign` queries. It first invokes the `Sign` algorithm to obtain a complete signature. Then, it can convert the complete signature into a pre-signature by y. Hence, it keeps that $\Pr[\mathsf{Game}_2 = 1] \leq \Pr[\mathsf{Game}_1 = 1] + \epsilon_2(n)$.

`Game 3`*:* We use the same changes made in `Game 1`. During the challenge phase, `Game 3` can extract a witness y, then check whether $(Y, y) \in R_Y$. If it does not hold, then the game outputs $\perp$.

Claim: Asuume Bad_2 is the event that `Game 3` outputs $\perp$, $\Pr[\mathsf{Bad}_2] \leq \epsilon_1(n)$.

Proof: As the *online extractor* property of NIZK, for the witness y extracted from (Y, π), it keeps that $(Y, y) \in R_Y$.

As the only difference is that event Bad_2 happens, it keeps that $\Pr[\mathsf{Game}_3 = 1] \leq \Pr[\mathsf{Game}_2 = 1] + \epsilon_1(n)$.

`Game 4`*:* When getting $\mathcal{A}$'s the challenge message m, this game invokes the `Sign` algorithm to obtain a complete signature, then transforms it into a pre-signature by y. As a result, we have the same indistinguishability argument in the `Game 2`, it keeps that $\Pr[\mathsf{Game}_4 = 1] \leq \Pr[\mathsf{Game}_3 = 1] + \epsilon_2(n)$.

Then, we need to show that there exists $\mathcal{S}$ which can perfectly simulate `Game 4` and uses $\mathcal{A}$ to win the `strongSigForge` game. The steps are as follows.

1) *Sign queries:* When $\mathcal{A}$ querying $\mathcal{O}_S$, $\mathcal{S}$ sends m to its oracle Sign^{P1363} and sends the result to $\mathcal{A}$.
2) *Random Oracle queries:* When $\mathcal{A}$ querying $\mathcal{H}(x)$, if $H[x] = \perp$, $\mathcal{S}$ queries $\mathcal{H}^{P1363}$, otherwise returns $H[x]$.
3) *Pre-Sign queries:*
a) When $\mathcal{A}$ querying $\mathcal{O}_{pS}$, the simulator $\mathcal{S}$ extracts the witness y using the *online extractor* property, sends m to the Sign^{P1363} oracle and parses σ as (h, S).
b) $\mathcal{S}$ computes a pre-signature $(h, \tilde{S}, \pi)$ by calculating $\tilde{S} := S - y$.
4) *Challenge phase:*
a) When $\mathcal{A}$ outputting the challenge message m^*, $\mathcal{S}$ creates (Y, y, π) by `GenR` algorithm, sends m^* to the Sign^{P1363} oracle and parses σ as (h, S).
b) $\mathcal{S}$ computes a pre-signature $(h, \tilde{S}, \pi)$ by calculating $\tilde{S} := S - y$.
c) When $\mathcal{A}$ outputting a forgery σ^*, $\mathcal{S}$ outputs (m^*, σ^*) as its forged message/signature pair.

TABLE 1 Experimental Results on `Miracl` Library

Symbol	Operation	Cost (ms)
T_{bp}	Bilinear pairing	11.20
T_{add1}	Point addition ($\mathbb{G}_1$)	0.02
T_{mul1}	Point multiplication ($\mathbb{G}_1$)	0.83
T_{add2}	Point addition ($\mathbb{G}_2$)	0.03
T_{mul2}	Point multiplication ($\mathbb{G}_2$)	4.27
T_{mul_T}	Multiplication operation ($\mathbb{G}_T$)	0.08
T_{exp}	Exponentiation operation ($\mathbb{G}_T$)	1.63
$T_{\mathcal{H}}$	Hash function	0.03
T_{add}	Modular addition ($\mathbb{Z}_q^*$)	0.01
T_{mul}	Modular multiplication ($\mathbb{Z}_q^*$)	0.01
T_{inv}	Modular inversion ($\mathbb{Z}_q^*$)	0.01

ms represents milliseconds.

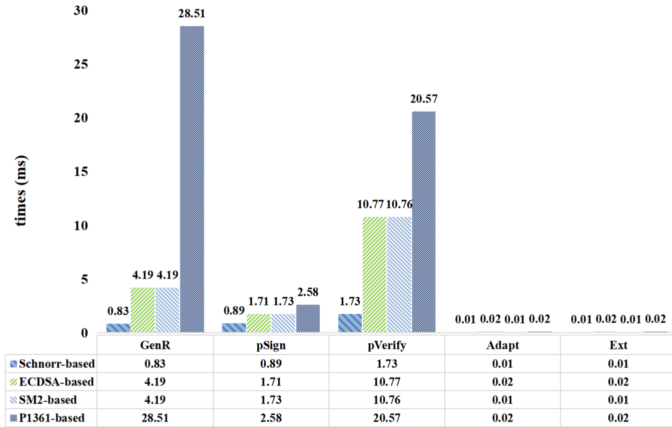


FIGURE 2. Execution time of different schemes.

The difference between the aforementioned simulation and `Game 4` is syntactical. Rather than using the `Sign` and $\mathcal{H}$ oracles, the simulator $\mathcal{S}$ uses Sign^{P1363} and $\mathcal{H}^{P1363}$. Finally, we need to prove that the simulator $\mathcal{S}$ can use (m^*, σ^*) to win in the `strongSigForge`.

Claim: (m^*, σ^*) is a valid forged message/signature pair in `strongSigForge`.

Proof: Before the challenge phase, $\mathcal{A}$ has not queried the $\mathcal{O}_S$ or $\mathcal{O}_{pS}$ on the message m^*. Thus, Sign^{P1363} is is only queried on m^*. As we discussed in `Game 0`, the probability of a event that the σ output by $\mathcal{A}$ is equal to the σ output by Sign^{P1363} is negligible. Thus, Sign^{P1363} has not output σ^* on a input m^*. Consequently, (m^*, σ^*) is a valid forged message/signature pair in `strongSigForge`.

Finally, we know the transformations from `Game 0` to `Game 4` is indistinguishable. We can get $\Pr[\text{aWinExt}_{\mathcal{A},\Xi_{R,\Pi_{Sig}}}(1^n) = 1] = \Pr[\mathsf{Game}_0 = 1] \le \Pr[\mathsf{Game}_4 = 1] + 2\epsilon_1(n) + 2\epsilon_2(n) \le \Pr[\mathsf{strongSigForge}_{\mathcal{A},\Pi_{Sig}} = 1] + 2\epsilon_1(n) + 2\epsilon_2(n)$. □

VII. EXPERIMENTAL FINDINGS

We analyze the computation as well as communication costs of $\Xi_{R,\Pi_{Sig}^{P1363}}$, and compare it with the costs of the existing `Schnorr`-AS, `ECDSA`-AS `SM2`-AS [7], [23]. The `Schnorr`-AS and `ECDSA`-AS schemes were proposed by Aumayr et al. in [7]. They gave the instantiations and detailed proofs. `SM2`-AS was proposed by Peng et al. in [23].

A. COMPUTATION COST

We show the computation cost imposed by the different phases, such as `pSign`, `pVerify`, `Adapt`, `Ext`. As the `Setup` and `Verify` phases in adaptor signatures are the same as the original signature, we ignore these. We compute different cryptographic operations on a personal computer utilizing `MIRACL` Library. We use Table 1 to demonstrate the computation cost of each operation.

Meanwhile, we use T_P^π and T_V^π to represent the costs of NIZK.`Prove` and NIZK.`Verify`. We compute theoretical computation complexity of each scheme, i.e., `Schnorr`-AS, `ECDSA`-AS and `SM2`-AS, as shown in Table 2. Note that when we calculate the cost of SM2 adaptor signatures, we adopt the optimized version with pre-computation. As for the zero-knowledge proofs used in each scheme, `ECDSA`-AS and `SM2`-AS use the Σ-protocol to prove them. We also use Σ-protocol to instantiate $\pi = \mathsf{Prove}(Y, y) : Y = \tilde{e}(y, h_{ID} \cdot Q_2 + P)\}$, see V-A. Note that we ignore the time cost of the operation $\tilde{e}(Q_1, C)$ where $C = h_{ID} \cdot Q_2 + P$, because this part can be pre-calculated. Further, we calculate the algorithm execution time of each scheme in Fig. 2 according to Table 1. Due to the use of bilinear pairs, it can be seen that the time costs `GenR` and `pVerify` of our scheme are relatively large than others.

B. COMMUNICATION COST

In this section, we use $|\mathbb{Z}_q^*|$ and $|\mathbb{G}_1|$ represent the element sizes in $\mathbb{Z}_q^*$ and $\mathbb{G}_1$. We set $|\mathbb{Z}_q^*| = 32$ bytes and $|\mathbb{G}_1| = 64$ bytes. For communication cost, we consider the size of pre-signatures, as the public key, private key and signature sizes are the same as the original schemes. The pre-signature size of `Schnorr`-AS is $2|\mathbb{Z}_q^*|$. The pre-signature sizes of `ECDSA`-AS and `SM2`-AS are $4|\mathbb{Z}_q^*| + |\mathbb{G}|$, as each zero-knowledge proof size is $2|\mathbb{Z}_q^*|$. As for our scheme, the pre-signature is $(h, \tilde{S}, \pi)$. As $h \in \mathbb{Z}_q^*$, $\tilde{S} \in \mathbb{G}_1$ and $\pi := (e, z)$, where $e \in \mathbb{Z}_q^*, z \in \mathbb{Z}_q^*$, the pre-signature size is $3|\mathbb{Z}_q^*| + |\mathbb{G}_1|$. The pre-signature size comparison is shown in Table 3. The pre-signature values of our scheme is higher than the Schnorr-AS scheme, but lower than ECDSA-AS and SM2-AS schemes.

VIII. APPLICATIONS IN BLOCKCHAIN

In this section, we introduce two applications of adaptor signature on the identity-based blockchain: atomic swap and payment channel network. Note that we require the underlying blockchain supports identity services, such as HIBEchain [16], ChainMaker [17]. Thus, our scheme requires users to

TABLE 2 Computation Costs of Different Schemes

Adaptor signatures	GenR	pSign	pVerify	Adapt	Ext
Schnorr-AS	T_{mul1}	$T_{add1} + T_{mul1} + T_{\mathcal{H}} + T_{mul}$	$2T_{add1} + 2T_{mul1} + T_{\mathcal{H}}$	T_{add}	T_{add}
ECDSA-AS	$T_P^{\pi} + T_{mul1}$	$2T_{mul1} + T_{\mathcal{H}} + T_{inv} + T_{mul}$	$T_V^{\pi} + T_{add1} + 2T_{mul1} + T_{\mathcal{H}} + T_{inv} + 2T_{mul}$	$T_{inv} + T_{mul}$	$T_{inv} + T_{mul}$
SM2-AS	$T_P^{\pi} + T_{mul1}$	$T_{add1} + 2T_{mul1} + T_{\mathcal{H}} + 2T_{mul}$	$T_V^{\pi} + T_{add1} + 2T_{mul1} + T_{\mathcal{H}} + 2T_{mul}$	T_{add}	T_{add}
Ours	$T_P^{\pi} + T_{bp} + T_{add2} + T_{mul2} + T_{\mathcal{H}}$	$T_{mul_T} + T_{exp} + T_{\mathcal{H}} + T_{add} + T_{mul}$	$T_V^{\pi} + T_{bp} + T_{add2} + T_{mul2} + T_{mul_T} + T_{exp} + T_{\mathcal{H}}$	T_{add1}	T_{add1}

TABLE 3 Communication Costs of Different Schemes

Schemes	Pre-signature sizes	Values
Schnorr-AS	$2\lvert\mathbb{Z}_q\rvert$	64 bytes
ECDSA-AS	$4\lvert\mathbb{Z}_q^*\rvert + \lvert\mathbb{G}_1\rvert$	192 bytes
SM2-AS	$4\lvert\mathbb{Z}_q^*\rvert + \lvert\mathbb{G}_1\rvert$	192 bytes
Ours	$3\lvert\mathbb{Z}_q^*\rvert + \lvert\mathbb{G}_1\rvert$	160 bytes

register first. Namely, users need to obtain their private keys from KGC.

A. ATOMIC SWAPS

Atomic swaps is a technology that supports the fast exchange of two cryptocurrencies running on different blockchain networks. Two users u_1 and u_2 can exchange two different cryptocurrencies c_1 and c_2 in a fair method. In Bitcoin, it is usually implemented by a hash-lock mechanism. While with an adaptor signature scheme, it is viable for any blockchain system that only needs to support digital signatures (e.g., Schnorr/ECDSA signatures). Note that signatures without randomness cannot be transformed into adaptor signatures [21]. Our proposed identity-based adaptor signature scheme is completely applicable to the atomic swaps on a consortium blockchain. It works as follows (also see Fig. 3):

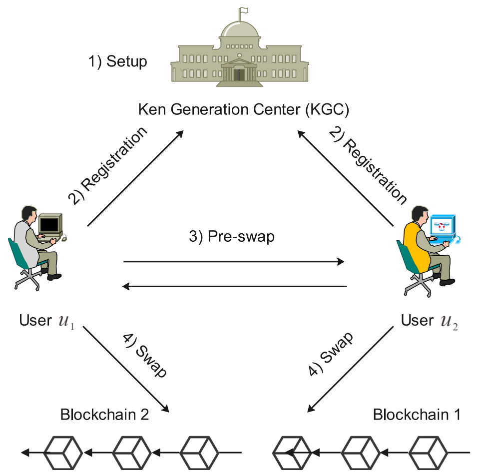


FIGURE 3. Atomic swaps with our proposed scheme.

1) *Setup:* First, we need run the $\Xi_{R,\Pi_{Sig}^{P1363}}$.Setup$(1^n)$ to generates the public parameter $\mathbb{PP}$. The KGC obtains its master secret key s and public key $P = s \cdot Q_2$. The user's ID is viewed as an address in our target blockchain.
2) *Registration:* The user obtains the private key d_{ID} by requiring KGC to execute $\Xi_{R,\Pi_{Sig}^{P1363}}$.Extract$(ID, s)$. At the end, users u_1 and u_2 get their private key $d_{ID}^{u_1}$ and $d_{ID}^{u_2}$, respectively.
3) *Pre-swap:* The user u_1 first runs $\Xi_{R,\Pi_{Sig}^{P1363}}$.GenR$(Y, y)$ to get a statement/witness pairs (Y, y) and the zero-knowledge proof π. Then, the user u_1 obtains $\tilde{\sigma}_1$ by running $\Xi_{R,\Pi_{Sig}^{P1363}}$.pSign$(tx_1, d_{ID}^{u_1}, Y, \pi)$ where tx_1 is a transaction spending the coins c_1 to u_2. The user u_1 sends $(tx_1, \tilde{\sigma}, Y)$ to the user u_2. After verifying the pre-signature by $\Xi_{R,\Pi_{Sig}^{P1363}}$.pVerify $(tx_1, \tilde{\sigma}, ID, Y)$, the user u_2 obtains $\tilde{\sigma}_2$ by running $\Xi_{R,\Pi_{Sig}^{P1363}}$.pSign$(tx_2, d_{ID}^{u_2}, Y, \pi)$ where tx_2 is a transaction spending the coins c_2 to u_1. Finally, the user u_2 sends $\tilde{\sigma}_2$ to u_1. Up to now, The two users have been successfully exchanged their pre-signatures $\tilde{\sigma}_1$ and $\tilde{\sigma}_2$.
4) *Swap:* The user u_1 runs $\Xi_{R,\Pi_{Sig}^{P1363}}$.Adapt$(\tilde{\sigma}_2, y)$ to get the transaction tx_2's complete signature σ_2. The user u_1 publishes (σ_2, tx_2) on the blockchain to gets c_2. Once seeing (σ_2, tx_2) on the blockchain, the user u_2 can run $\Xi_{R,\Pi_{Sig}^{P1363}}$.Ext$(\sigma_2, \tilde{\sigma}_2, Y)$ to get y, thereby obtaining the complete signature σ_1 by $\Xi_{R,\Pi_{Sig}^{P1363}}$.Adapt$(\tilde{\sigma}_1, y)$. Finally, the user u_2 publishes (σ_1, tx_1) on the blockchain to gets c_1.

B. PAYMENT CHANNEL NETWORKS

To solve the blockchain's scalability issue, payment channel networks (PCN) are frequently used, which is considered as the key solution of Layer 2. PCN allows users to establish off-chain payment channels, where they can conduct multiple private and nearly instant transactions without involving the blockchain. Only the final channel state is recorded on the blockchain when the channel is closed. By reducing on-chain

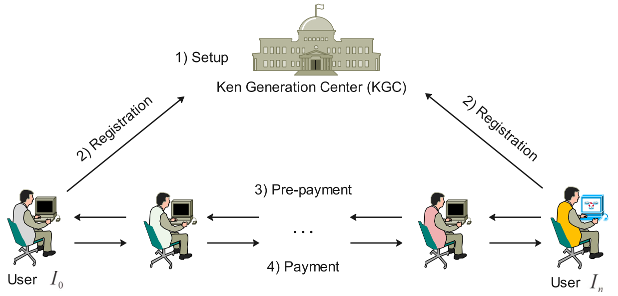


FIGURE 4. PCN with our proposed scheme.

transactions, PCN improves scalability and transaction efficiency on the blockchain network.

Specifically, a payment channel enables two users to make instant and arbitrarily multi-transactions between them and only the create and close phases needs to be on-chain, while the update phase for these arbitrary transactions can be off-chain. Furthermore, in PCN, when a sender S wants to transfer some coins to a receiver R (or I_n) without a direct connection, the sender S (or I_0) can do it through several intermediary nodes $I_1, \ldots, I_{n-1}$ as long as each channel has enough balance. We use AMHL method proposed in [15] to build PCN. It works as follows (also see Fig. 4):

1) *Setup and Registration:* The phases are the same as that in atomic swaps VIII-A, we ignore it here.
2) *Pre-payment:* First, the sender S first randomly chooses $(r_0, r_1, \ldots, r_{n-1}) \in_R \mathbb{Z}_q^*$, and computes $y_j = \sum_{j=0}^{j} r_j$, $Y_j = \mathsf{G}(y_j)$, where $j \in \{0, 1, \ldots, n-1\}$, G is an additively homomorphic one-way function. Second, the sender S runs $\Xi_{R,\Pi_{Sig}^{P1363}}$.GenR$(Y_j, y_j)$ to get the witness π_j for $j \in \{0, 1, \ldots, n-1\}$. Third, the sender S sends each tuple $(Y_{j-1}, Y_j, r_j, \pi_j)$ to each intermediary I_j where $j \in \{1, \ldots, n-1\}$ and sends (Y_{n-1}, y_{n-1}) to the receiver R. Each intermediary I_j can check whether $\mathsf{G}(r_j) \oplus Y_{j-1} = Y_j$ is correct, where $\oplus$ is the homomorphic addition.
3) *Payment:* First, the sender S runs $\Xi_{R,\Pi_{Sig}^{P1363}}$.pSign $(tx_0, d_{ID}^{I_0}, Y_0, \pi_0)$ to get a pre-signature $\tilde{\sigma}_0$ where tx_0 is a transaction spending the coins S (or I_0) to I_1 and sends $\tilde{\sigma}$ to I_1. Secnod, for $j = \{1, 2, \ldots, n-1\}$, I_j makes the similar pre-signature $\tilde{\sigma}_j$ with a condition on preimage of Y_{j+1}. Third, for $j = \{0, 1, 2, \ldots, n-1\}$, I_{j+1} checks whether $\Xi_{R,\Pi_{Sig}^{P1363}}$.pVerify$(tx_j, \tilde{\sigma}_j, ID_j, Y_j)$ is correct. Once all conditional payments are done, the sender S uses y_{n-1} to run $\Xi_{R,\Pi_{Sig}^{P1363}}$.Adapt$(\tilde{\sigma}_{n-1}, y_{n-1})$ to obtain the signature σ_{n-1} on tx_{n-1}, thereby redeeming the I_{n-1}'s transferred coins. S sends σ_{n-1} to I_{n-1}. I_{n-1} then runs $\Xi_{R,\Pi_{Sig}^{P1363}}$.Ext$(\sigma_{n-1}, \tilde{\sigma}_{n-1}, Y_{n-1})$ to get y_{n-1} and computes $y_{n-2} = y_{n-1}/l_{n-1}$. Later, I_{n-1} can use y_{n-2} to get the signature σ_{n-2} on tx_{n-2} for transferring coins from I_{n-2}. And so on, this process continues until I_1 obtains R's coins.

IX. CONCLUSION

In this article, we propose an adaptor signature scheme for the `IEEE P1363 standard`. We formally prove that our scheme meets the desired properties. Findings from the evaluation show that our scheme is more expensive than others, but we emphasize that it is due to the defects of the identity based cryptosystem itself, e.g., the need to use bilinear pairs. We also present two applications for our signature, namely atomic swap and payment channel networks. In the future, we will consider designing lightweight identity-based adaptor signatures.

REFERENCES

[1] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, 2019.

[2] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, Sep./Oct. 2019.

[3] K. Croman et al., "On scaling decentralized blockchains: (A position paper)," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2016, pp. 106–125.

[4] L. Gudgeon, P. Moreno-Sanchez, S. Roos, P. McCorry, and A. Gervais, "SoK: Layer-two blockchain protocols," in *Proc. 24th Int. Conf. Financial Cryptography Data Secur.*, 2020, pp. 201–226.

[5] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Pro. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 455–471.

[6] P. Moreno-Sanchez, A. Blue, D. V. Le, S. Noether, B. Goodell, and A. Kate, "DLSAG: Non-interactive refund transactions for interoperable payment channels in Monero," in *Proc. 24th Int. Conf. Financial Cryptography Data Secur.*, 2020, pp. 325–345.

[7] L. Aumayr et al., "Generalized channels from limited blockchain scripts and adaptor signatures," in *Proc. 27th Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2021, pp. 635–664.

[8] E. Tairi, P. Moreno-Sanchez, and M. Maffei, "A2L: Anonymous atomic locks for scalability in payment channel hubs," in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 1834–1851.

[9] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2014.

[10] A. Poelstra, "Scriptless scripts," 2017. Accessed: Jan. 7, 2023. [Online]. Available: http://diyhpl.us/wiki/transcripts/layer2-summit/2018/scriptless-scripts/

[11] C. Decker and R. Wattenhofer, "A fast and scalable payment network with Bitcoin duplex micropayment channels," in *Proc. 17th Symp. Self-Stabilizing Syst.*, 2015, pp. 3–18.

[12] R. Yu, G. Xue, V. T. Kilari, D. Yang, and J. Tang, "CoinExpress: A fast payment routing mechanism in blockchain-based payment channel networks," in *Proc. IEEE 27th Int. Conf. Comput. Commun. Netw.*, 2018, pp. 1–9.

[13] M. Herlihy, "Atomic cross-chain swaps," in *Proc. ACM Symp. Princ. Distrib. Comput.*, 2018, pp. 245–254.

[14] A. Deshpande and M. Herlihy, "Privacy-preserving cross-chain atomic swaps," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2020, pp. 540–549.

[15] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.

[16] Z. Wan, W. Liu, and H. Cui, "HIBEChain: A hierarchical identity-based blockchain system for large-scale IoT," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1286–1301, Mar./Apr. 2023.

[17] C. E. Alliance, "Chainmaker," 2021. Accessed: Jan. 7, 2023. [Online]. Available: https://chainmaker.org.cn/home

[18] X. Guo, Q. Guo, M. Liu, Y. Wang, Y. Ma, and B. Yang, "A certificateless consortium blockchain for IoTs," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 496–506.

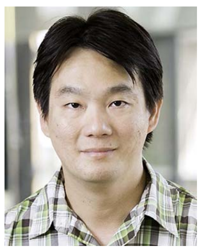
[19] P. S. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, “Efficient and provably-secure identity-based signatures and signcryption from bilinear maps,” in *Proc. 11th Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2005, pp. 515–532.
[20] L. Fournier, “One-time verifiably encrypted signatures AKA adaptor signatures,” 2019. Accessed: Jan. 7, 2023. [Online]. Available: https://github.com/LLFourn/one-time-VES/blob/master/main.pdf
[21] A. Erwig, S. Faust, K. Hostáková, M. Maitra, and S. Riahi, “Two-party adaptor signatures from identification schemes,” in *Proc. IACR Int. Conf. Public-Key Cryptography*, 2021, pp. 451–480.
[22] S. A. K. Thyagarajan and G. Malavolta, “Lockable signatures for blockchains: Scriptless scripts for all signatures,” in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 937–954.
[23] C. Peng, M. Luo, D. He, and X. Huang, “Adaptor signature scheme based on the SM2 digital signature algorithm,” *J. Comput. Res. Develop.*, vol. 58, no. 10, pp. 2278–2286, 2021.
[24] M. F. Esgin, O. Ersoy, and Z. Erkin, “Post-quantum adaptor signatures and payment channel networks,” in *Proc. Eur. Symp. Res. Comput. Secur.*, 2020, pp. 378–397.
[25] E. Tairi, P. Moreno-Sanchez, and M. Maffei, “Post-quantum adaptor signature for privacy-preserving off-chain payments,” in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2021, pp. 131–150.
[26] A. Menezes, “An introduction to pairing-based cryptography,” *Recent Trends Cryptography*, vol. 477, pp. 47–65, 2009.
[27] J. Katz, “Digital signatures: Background and definitions,” in *Digital Signatures*. Cham, Switzerland:Springer, 2010, pp. 3–33.
[28] M. Blum, P. Feldman, and S. Micali, “Non-interactive zero-knowledge and its applications,” in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. San Rafael, CA, USA: Morgan & Claypool, 2019, pp. 329–349.
[29] M. Fischlin, “Communication-efficient non-interactive proofs of knowledge with online extractors,” in *Proc. Annu. Int. Cryptology Conf.*, 2005, pp. 152–168.
[30] V. Shoup, “Sequences of Games: A tool for taming complexity in security proofs,” Cryptology Eprint Arch., Paper 2004/332, 2004. [Online]. Available: http://eprint.iacr.org/2004/332

ZIJIAN BAO received the M.S. degree in computer application technology from the School of Computer Science and Engineering, Northeastern University, Shenyang, China, in 2019. He is currently working toward the Ph.D. degree with the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His research focuses on cryptographic protocols.

DEBIAO HE (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. He has authored or coauthored more than 100 research papers in refereed international journals and conferences, such as the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and Usenix Security Symposium. His main research interests include cryptography and information security, in particular cryptographic protocols. Prof. He was the recipient of the 2018 IEEE Systems Journal Best Paper Award and 2019 IET Information Security Best Paper Award. His work has been cited more than 10000 times at Google Scholar. He is on the editorial board of several international journals, such as the ACM *Distributed Ledger Technologies: Research and Practice*, *Frontiers of Computer Science*, and IEEE TRANSACTIONS ON COMPUTERS.

CONG PENG received the Ph.D. degree in applied mathematics in 2021 from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, where he is currently an Associate Professor with the School of Cyber Science and Engineering. His main research interests include applied cryptography and data security.

MIN LUO received the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2003. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. He has authored or coauthored more than 50 research papers in refereed international journals and conferences, such as IEEE Symposium on Security and Privacy and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. His main research interests include cryptography and information security and blockchain security.



KIM-KWANG RAYMOND CHOO (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Brisbane, QLD, Australia, in 2006. He currently holds the Cloud Technology Endowed Professorship with The University of Texas at San Antonio, San Antonio, TX, USA. He is the founding Co-Editor-in-Chief of ACM Distributed Ledger Technologies: Research & Practice, and the founding Chair of IEEE Technology and Engineering Management Society Technical Committee (TC) on Blockchain and Distributed Ledger Technologies. He was the recipient of the 2022 IEEE Hyper-Intelligence TC Award for Excellence in Hyper-Intelligence Systems (Technical Achievement Award), 2022 IEEE TC on Homeland Security Research and Innovation Award, 2022 IEEE TC on Secure and Dependable Measurement Mid-Career Award, and 2019 IEEE TC on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher).