EFFICIENT IDENTITY-BASED CHAMELEON HASH FOR MOBILE DEVICES

Cong Li[⋆], *Qingni Shen*[⋆,†], *Zhikang Xie*[⋆], *Jisheng Dong*[⋆], *Yuejian Fang*[⋆], *Zhonghai Wu*[⋆]

[⋆] Peking University

ABSTRACT

Online/offline identity-based signature (OO-IBS) is an adequate cryptographic tool to provide the message authentication and integrity in mobile devices, since it lightens the computational burden after the signer receives the message and eliminates the overhead of certificate management. It has several valuable applications, such as wireless sensor networks and automatic dependent surveillance-broadcast systems. Identity-based chameleon hash (IB-CH), as an alternative building block to construct OO-IBS, has been explored in several literatures. Nevertheless, almost all of the prior IB-CH schemes are in the random oracle model, which may lead to security risks in practicality. The only IB-CH scheme in the standard model proposed by Xie et al. (ICC'21) suffers from the large size of public parameters and inefficient setup process. In this paper, we propose an efficient IB-CH scheme in the standard model, significantly reducing the computational costs of all the algorithms and the size of public parameters compared with Xie's scheme. The security and experimental analyses demonstrate the security and good performance of our scheme. Furthermore, we applied our scheme to optimize the existing generic OO-IBS construction. Our optimized construction reduces computational overhead by 50.0% in the online phase compared with the original construction.

Index Terms— identity-based chameleon hash, online/offline identity-based signature, mobile devices, standard model

1. INTRODUCTION

Nowadays, since mobile devices have become an indispensable part of our daily life, it is more and more necessary to protect sensitive information in them. Identity-based signatures (IBS) [1] as a fundamental cryptographic primitive is one of useful techniques to protect mobile devices. It eliminates the need of a certificate distribution architecture and is able to provide message authentication, message integrity, etc. Nevertheless, considering the limited computational power and battery capacity of mobile devices, numerous conventional IBS schemes are not appropriate for them due to the heavy computations required in their signing process. The online/offline IBS (OO-IBS) [2] is an alternative cryptographic tool to tackle this challenge. It splits the signature generation algorithm into two phases, the offline phase and the online phase. In the offline phase, the vast majority of work is done before the signer knows the message to be signed. While in the online phase, only very few operations need to be conducted after the signer learns the message. Thus the signer can finish the signing process quickly.

Shamir and Tauman [3] first employed the chameleon hash (CH) [4], which is a trapdoor collision-resistant hash function (the trapdoor holder can find collisions efficiently) and has been studied by numerous literatures [5–12], to design a generic transformation to convert digital signatures to online/offline signatures. Recently, this idea was transplanted to build OO-IBS by the literature [13] using the identity-based chameleon hash (IB-CH) (the notion of IB-CH was proposed in the literature [14]), which employs the user's identity instead of the hash key to calculate hash values. For this generic transformation shown in Figure 1, in the offline phase, the signer first computes a hash value h (of an IB-CH) with a random message R and a randomness r as inputs. Then he signs h to generate the signature σ with the underlying IBS scheme. Later in the online phase, after the signer learns the message m to be signed, he merely needs to find a collision of h by calculating a fresh randomness r' with the message m, the randomness r, the random message R and his trapdoor td as inputs. With the randomness r', h is the hash value of m. That is, the message m and the random message R have the same hash value, and the input of underlying IBS remains unchanged. Hence, σ is a valid signature of the message m.

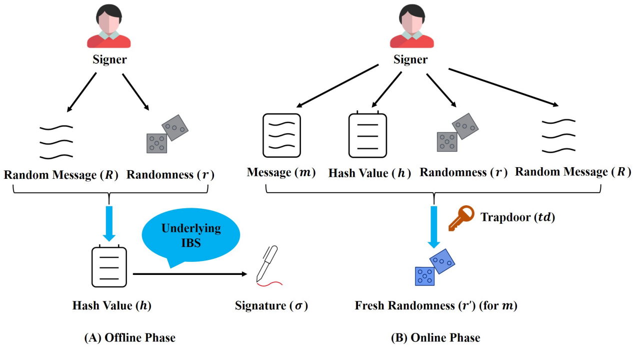

Fig. 1. Generic transformation of OO-IBS based on IB-CH

This work was supported by National Natural Science Foundation of China (No. 61672062, 61232005). Qingni Shen[†] is the corresponding author.

Obviously, IB-CH is a critical component of this generic transformation, and deeply affects the performance and security of it. Nevertheless, nearly all of the existing IB-CH schemes [14–19] are in the random oracle model, which has been pointed out to have defects in practicality [20]. Moreover, the only existing IB-CH scheme [13] in the standard model suffers from the large size of public parameters and inefficient setup process.

To tackle the aforementioned problem, in this paper, we focus on designing a novel IB-CH scheme in the standard model with better performance. The contributions can be summarized as follows:

(1) We propose an efficient IB-CH scheme for mobile devices based on the Gentry signature [21,22], achieving adaptive identity, basic collision resistance (ID-B-CollRes) in the standard model. Then we provide a formal security analysis to demonstrate the security of our scheme.

(2) We firstly implement our IB-CH scheme with previous IB-CH ones [13,15] as benchmarks. Then we give theoretical and experimental analyses of ours and those IB-CH schemes in computational and storage costs. Both of the analyses indicate that our IB-CH scheme makes significant improvements in efficiency and is suitable for mobile devices, especially the experimental one illustrating that our scheme reduces running time by approximately 73.5%, 55.0%, 51.4% and 53.0% in Setup, KeyGen, Hash and Col algorithms respectively, and reduces the size of public parameters by about 98.1% compared with scheme [13] when the length of identity is 256 bits.

(3) We apply our IB-CH scheme to optimize the existing generic OO-IBS construction [13]. In comparison with it, our optimized construction reduces computational overhead by 50.0% in the online phase and costs approximately two pairing operations less in the verifying process.

2. PRELIMINARIES

Definition 1 (Identity-based chameleon hash). *Our IB-CH scheme consists of the following four algorithms:*

- $\mathsf{Setup}(1^\lambda) \rightarrow (pp, msk)$: *The setup algorithm takes as input a security parameter λ. It returns the public parameters pp and the master secret key msk. For simplicity, the public parameters pp are taken as input implicitly in the descriptions of following algorithms.*
- $\mathsf{KeyGen}(msk, ID) \rightarrow td_{ID}$: *The key generation algorithm takes as inputs the master secret key msk and an identity ID. It generates a trapdoor td_{ID} related to ID.*
- $\mathsf{Hash}(ID, m) \rightarrow (h, r)$: *The hash calculation algorithm takes as inputs an identity ID and a message m. It outputs a hash/randomness pair (h, r).*
- $\mathsf{Col}(td_{ID}, h, m, r, m') \rightarrow r'$: *The collision finding algorithm takes as inputs a trapdoor td_{ID}, a hash value h, a message m, a randomness r and a fresh message m'. It outputs a new randomness r' such that* $\mathsf{Hash}(ID, m; r) = \mathsf{Hash}(ID, m'; r') = h$.

To verify the validity of a triple (h, m, r) under an identity ID, the verification operation can simply check whether the equation $\mathsf{Hash}(ID, m; r) = h$ holds.

Definition 2 (ID-B-CollRes [13]). *An identity-based chameleon hash $\mathcal{IB}\text{-}\mathcal{CH} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Hash}, \mathsf{Col})$ is said to be adaptive identity, basic collision resistant (ID-B-CollRes) if for any probabilistic polynomial-time adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible in the security parameter λ. The advantage of $\mathcal{A}$ is defined as*

$$\begin{aligned}\mathsf{Adv}^{\text{ID-B-CollRes}}_{\mathcal{A},\mathcal{IB}\text{-}\mathcal{CH}}(\lambda) = \Pr[&(ID^*, h^*, m^*, r^*, m'^*, r'^*) \\ &\leftarrow \mathsf{Exp}^{\text{ID-B-CollRes}}_{\mathcal{A},\mathcal{IB}\text{-}\mathcal{CH}}(1^\lambda) : \mathsf{Collide}].\end{aligned}$$

The $\mathsf{Exp}^{\text{ID-B-CollRes}}_{\mathcal{A},\mathcal{IB}\text{-}\mathcal{CH}}$ *experiment is defined in Figure 2 and the* $\mathsf{Collide}$ *event is defined as*

$$\mathsf{Hash}(ID^*, m^*; r^*) = \mathsf{Hash}(ID^*, m'^*; r'^*) = h^* \wedge \\ m^* \neq m'^* \wedge ID^* \notin \mathcal{H}.$$

```
$\mathsf{Exp}^{\text{ID-B-CollRes}}_{\mathcal{A},\mathcal{IB}\text{-}\mathcal{CH}}(1^\lambda)$
  $(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$
  $\mathcal{H} \leftarrow \varnothing$
  $(ID^*, h^*, m^*, r^*, m'^*, r'^*) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{KeyGen}}_{msk}(\cdot)}(pp)$
    where $\mathcal{O}^{\mathsf{KeyGen}}_{msk}(\cdot)$ on input $ID$:
      $td_{ID} \leftarrow \mathsf{KeyGen}(msk, ID)$
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{ID\}$
      return $td_{ID}$
```

Fig. 2. The $\mathsf{Exp}^{\text{ID-B-CollRes}}_{\mathcal{A},\mathcal{IB}\text{-}\mathcal{CH}}$ experiment

Definition 3 (Semantic security). *An identity-based chameleon hash $\mathcal{IB}\text{-}\mathcal{CH}$ is said to be semantically secure, if for all identity ID and all of the pairs of messages m and m', the probability distributions of the random variables* $\mathsf{Hash}(ID, m; r)$ *and* $\mathsf{Hash}(ID, m'; r)$ *are computationally indistinguishable.*

3. PROPOSED EFFICIENT IB-CH

In this section, we firstly describe the construction of our IB-CH, and then demonstrate its correctness.

Our IB-CH construction works as follows.

- $\mathsf{Setup}(1^\lambda) \rightarrow (pp, msk)$: The setup algorithm first calls the group generator algorithm $\mathcal{G}(1^\lambda)$ to obtain the descriptions of the groups and the bilinear map $D = (p, \mathbb{G}, \mathbb{G}_T, g, e)$. Then it randomly chooses $\alpha, \beta \in \mathbb{Z}_p$ and computes $g_1 = g^\alpha, g_2 = g^\beta, e(g, g), e(g_2, g)$. Finally, it outputs the public parameters pp and the master secret key msk as

$$pp = (D, g_1, g_2, e(g, g), e(g_2, g)),\ msk = (\alpha, \beta).$$

- $\mathsf{KeyGen}(msk, ID) \rightarrow td_{ID}$: To generate the trapdoor td_{ID} related to an identity $ID \in \mathbb{Z}_p^*$, the trapdoor generation algorithm randomly picks $t \in \mathbb{Z}_p$ and outputs the trapdoor td_{ID} as

$$td_{ID} = (td_1, td_2) = (t, g^{\frac{\beta - t}{\alpha - ID}}).$$

• $\mathsf{Hash}(\boldsymbol{ID}, \boldsymbol{m}) \rightarrow (\boldsymbol{h}, \boldsymbol{r})$: To compute the hash value of a message $m \in \mathbb{Z}_p$ associated with an identity $ID \in \mathbb{Z}_p^*$, the hash calculation algorithm randomly selects $r_1 \in \mathbb{Z}_p, r_2 \in \mathbb{G}$ and outputs the hash/randomness pair (h, r) as

$$h = e(g_2, g)^m e(g, g)^{r_1} e(r_2, g_1 g^{-ID}), r = (r_1, r_2).$$

• $\mathsf{Col}(\boldsymbol{td_{ID}}, \boldsymbol{h}, \boldsymbol{m}, \boldsymbol{r}, \boldsymbol{m'}) \rightarrow \boldsymbol{r'}$: The collision finding algorithm first checks whether the equation $\mathsf{Hash}(ID, m; r) = h$ holds. If not, it outputs $\perp$; otherwise it outputs a randomness r' for the fresh message m' to yield h as

$$r' = (r_1', r_2') = (r_1 + (m - m')td_1, r_2 \cdot td_2^{m-m'}).$$

– Correctness: Fristly, we have

$$\begin{aligned}
&e(g,g)^{td_1} e(td_2, g_1 g^{-ID}) = e(g,g)^t e(g^{\frac{\beta-t}{\alpha-ID}}, g_1 g^{-ID}) \\
&= e(g,g)^t e(g^{\frac{\beta-t}{\alpha-ID}}, g^{\alpha-ID}) = e(g,g)^t e(g,g)^{\beta-t} \\
&= e(g,g)^\beta = e(g_2, g).
\end{aligned}$$

Then if $\mathsf{Setup}(1^\lambda) \rightarrow (pp, msk)$, $\mathsf{KeyGen}(msk, ID) \rightarrow td_{ID}$, $\mathsf{Hash}(ID, m) \rightarrow (h, r)$, $\mathsf{Col}(td_{ID}, h, m, r, m') \rightarrow r'$, the following expressions hold.

$$\begin{aligned}
&\mathsf{Hash}(ID, m'; r') = e(g_2, g)^{m'} e(g,g)^{r_1'} e(r_2', g_1 g^{-ID}) \\
&= e(g_2, g)^{m'} e(g,g)^{r_1 + (m-m')td_1} e(r_2 \cdot td_2^{m-m'}, g_1 g^{-ID}) \\
&= e(g_2, g)^{m'} e(g,g)^{r_1} e(g,g)^{(m-m')td_1} \cdot e(r_2, g_1 g^{-ID}) \\
&\quad \cdot e(td_2^{m-m'}, g_1 g^{-ID}) \\
&= e(g_2, g)^{m'} (e(g,g)^{td_1} e(td_2, g_1 g^{-ID}))^{m-m'} \\
&\quad \cdot e(g,g)^{r_1} e(r_2, g_1 g^{-ID}) \\
&= e(g_2, g)^{m'} e(g_2, g)^{m-m'} e(g,g)^{r_1} e(r_2, g_1 g^{-ID}) \\
&= e(g_2, g)^m e(g,g)^{r_1} e(r_2, g_1 g^{-ID}) = \mathsf{Hash}(ID, m; r)
\end{aligned}$$

4. SECURITY ANALYSIS

We present a brief security analysis of our IB-CH scheme here.

Theorem 1. *If the q-SDH assumption holds in $\mathbb{G}$, our IB-CH is ID-B-CollRes secure in the standard model.*

Proof. If the q-SDH assumption [23] holds in $\mathbb{G}$, the Gentry signature scheme [21, 22] is existentially unforgeable against adaptive chosen-message attacks (EU-CMA). Assume there exists a PPT adversary $\mathcal{A}$ who can break our IB-CH scheme with non-negligible advantage ε. Then using this adversary, we can construct a simulator $\mathcal{B}$ to break the Gentry signature scheme [21, 22] (also called the underlying signature) with non-negligible advantage. The Gentry signature scheme [21, 22] and our IB-CH scheme are denoted as $\mathcal{SIG} = \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver}\}$ and $\mathcal{IB}\text{-}\mathcal{CH}$ in the following proof, respectively.

Setup. $\mathcal{SIG}$ gives $D = (p, \mathbb{G}, \mathbb{G}_T, g, e)$ and $pk_{\mathcal{SIG}} = (g_1, g_2)$ to the simulator $\mathcal{B}$. $\mathcal{B}$ computes $e(g,g), e(g_2, g)$ and sets the public parameters $pp = (D, g_1, g_2, e(g,g), e(g_2, g))$. Then $\mathcal{B}$ sends pp to $\mathcal{A}$. Notice that, the master secret key $msk = (\alpha, \beta)$ of $\mathcal{IB}\text{-}\mathcal{CH}$ is unknown to $\mathcal{B}$.

Trapdoor Query. The adversary $\mathcal{A}$ issues trapdoor queries in this phase. For each trapdoor query on ID, $\mathcal{B}$ submits ID to $\mathcal{SIG}$ and obtains the corresponding signature σ_{ID} (with respect to $pk_{\mathcal{SIG}}$) as

$$\sigma_{ID} = (\sigma_1, \sigma_2) = (t, g^{\frac{\beta-t}{\alpha-ID}}),$$

where $t \in \mathbb{Z}_p$. Eventually, $\mathcal{B}$ sets the trapdoor $td_{ID} = (td_1, td_2) = (\sigma_1, \sigma_2)$ and sends td_{ID} to $\mathcal{A}$.

Collide. The adversary $\mathcal{A}$ outputs a collision on an identity ID^* (trapdoors related to ID^* have never been queried before). Suppose the collision consists of (h, m, r) and (h, m', r'), where $m \neq m', r = (r_1, r_2)$ and $r' = (r_1', r_2')$, then the following equation holds.

$$\begin{aligned}
&e(g_2, g)^m e(g,g)^{r_1} e(r_2, g_1 g^{-ID^*}) \\
=&e(g_2, g)^{m'} e(g,g)^{r_1'} e(r_2', g_1 g^{-ID^*})
\end{aligned} \tag{1}$$

According to Equation (1), we have

$$e((r_2'/r_2)^{\frac{1}{m-m'}}, g_1 g^{-ID^*}) = e(g_2 g^{-(\frac{r_1'-r_1}{m-m'})}, g). \tag{2}$$

We let $\sigma_{ID^*}^* = (\sigma_1^* = \frac{r_1'-r_1}{m-m'}, \sigma_2^* = (r_2'/r_2)^{\frac{1}{m-m'}})$, then Equation (2) can be re-written as

$$e(\sigma_2^*, g_1 g^{-ID^*}) = e(g_2 g^{-\sigma_1^*}, g).$$

Namely, $\mathcal{SIG}.\mathsf{Ver}(ID^*, \sigma_{ID^*}^*) = 1$. Meanwhile, $\mathcal{B}$ has never submitted ID^* to $\mathcal{SIG}$. Consequently, $\mathcal{B}$ successfully forges a valid signature $\sigma_{ID^*}^*$ on the message ID^* in $\mathcal{SIG}$, and then can utilize $\sigma_{ID^*}^*$ to break $\mathcal{SIG}$.

Considering the simulation does not abort, we have $\mathsf{Adv}^{\text{EU-CMA}}_{\mathcal{B},\mathcal{SIG}} = \mathsf{Adv}^{\text{ID-B-CollRes}}_{\mathcal{A},\mathcal{IB}\text{-}\mathcal{CH}} = \varepsilon$. This completes the proof of Theorem 1. □

Theorem 2. *Our IB-CH is semantically secure.*

Due to space constraints, we omit the detailed proof of Theorem 2 here. It can be proved using the analogous method for the proof of the semantic security of scheme [13].

5. PERFORMANCE EVALUATION

In this section, we compare the computational and storage costs of our proposed scheme with previous IB-CH schemes [13, 15] in the theoretical and experimental aspects.

Implementation and Setup. In our implementation[1], we use

[1] The source code has been published in GitHub and the address is https://github.com/cleverli2008/Snowman-J/IB-CH.

the Java Pairing Based Cryptography Library [24] (Version 2.0.0) and select the Type A pairings built on the elliptic curve $y^2 = x^3 + x$ over the field $\mathbb{F}_q$ with embedding degree 2 for some suitable prime p ($|p| = 160$ bits). Then we set the length of identity to be 256 bits. The experiments are conducted on a desktop computer with Intel Core(TM) i7-3770 (3.4GHz * 4) and 16GB RAM running Windows 10 Pro 64-bit (Version 10.0.17134.407) and JDK 16.0. In addition, for each algorithm, we sample 30 times and take the average running time.

Performance. For the computational cost, the Table 1 illustrates that our scheme is superior to the other IB-CH schemes [13,15] in all the algorithms except Setup. Compared with the only existing IB-CH scheme [13] in the standard model, ours costs E, $2P + R - (E + E_T)$ and E less in KeyGen, Hash and Col, respectively. Besides, in Setup, ours costs $2P + E + 2R$ more than both of the schemes in the literature [15] do. But considering Setup is a one-time operation and usually run by the private key generator (PKG) server, the increased cost is acceptable. The results of experiments also demonstrate the good performance of our scheme. As shown in Table 3, in comparison with the scheme [13], ours reduces running time by approximately 73.5%, 55.0%, 51.4% and 53.1% in Setup, KeyGen, Hash and Col respectively, which is a significant improvement in efficiency. For the storage overhead, as illustrated in Table 2 and Table 3, although the schemes [15] in the random oracle model are better than ours, the gap is relatively small. Meanwhile, compared to the scheme [13], our scheme is still prior to it, especially reducing the size of public parameters by about 98.1%. In summary, our IB-CH scheme is efficient and secure in the standard model simultaneously, making it suitable for mobile devices.

Table 1. Comparison among IB-CH schemes on the theoretical computational cost

Schemes	Setup	KeyGen	Hash	Col
Scheme1 [15]	$PG + E$	$E + H_G$	$2P + E + H_G + R$	E
Scheme2 [15]	$PG + E$	$E + H_{Z_p}$	$2P + E + 2E_T + H_{Z_p} + R$	$3E$
[13]	$PG + 2E + (n+2)R$	$2E$	$3P + E_T + 2R$	$2E$
Ours	$PG + 2P + 2E + 2R$	E	$P + E + 2E_T + R$	E

PG: the group generation operation; E, E_T: the exponentiation operations in $\mathbb{G}$ and $\mathbb{G}_T$, respectively; P: the pairing operation, R: the randomness generation operation in $\mathbb{G}$; H_G, H_{Z_p}: the hash calculation operations with $\mathbb{G}$ as the output space and $\mathbb{Z}_p$ as the output space, respectively; n: the length of identity employed in the scheme [13].

Table 2. Comparison among IB-CH schemes on the theoretical storage cost

Schemes	$\|pp\|$	$\|td\|$	$\|h\|$	$\|r\|$
Scheme1 [15]	$2\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|\mathbb{G}_T\|$	$\|\mathbb{G}\|$
Scheme2 [15]	$2\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|\mathbb{G}_T\|$	$\|\mathbb{G}\|$
[13]	$(n+4)\|\mathbb{G}\|$	$2\|\mathbb{G}\|$	$\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|$
Ours	$3\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$\|\mathbb{Z}_p\| + \|\mathbb{G}\|$	$\|\mathbb{G}_T\|$	$\|\mathbb{Z}_p\| + \|\mathbb{G}\|$

$|pp|, |td|, |h|, |r|$: the sizes of the public parameters pp, a trapdoor td, a hash value h and the randomness r, respectively; $|\mathbb{G}|, |\mathbb{G}_T|$: the element lengths in $\mathbb{G}$ and $\mathbb{G}_T$, respectively; $|\mathbb{Z}_p|$: the element length in $\mathbb{Z}_p$; n: the same as that in Table 1.

Table 3. Comparison among IB-CH schemes on running time of various algorithms and sizes of various components

Schemes	Scheme1 [15]	Scheme2 [15]	[13]	Ours
Setup	103.646	103.420	558.670	148.322
KeyGen	53.361	14.537	30.279	13.636
Hash	61.297	43.373	48.226	23.442
Col	10.871	34.751	23.767	11.155
$\|pp\|$	0.634	0.634	81.354	1.577
$\|td\|$	0.309	0.308	0.625	0.368
$\|h\|$	0.311	0.312	0.312	0.312
$\|r\|$	0.308	0.308	0.623	0.365

$|pp|, |td|, |h|, |r|$: the same as those in Table 2; The unit of running time for Setup, KeyGen, Hash, Col is millisecond (ms); The unit of file'size for $|pp|, |td|, |h|, |r|$ is kibibyte (KB).

6. OPTIMIZE THE GENERIC ONLINE/OFFLINE IBS CONSTRUCTION

The literature [13] first formalizes a generic transformation to convert any IBS construction with EU-CMA security to an OO-IBS construction with EU-CMA security based on their IB-CH scheme. Using our scheme as the underlying IB-CH, we can optimize their generic OO-IBS construction. As shown in Figure 1, the generic transformation merely runs Col of the underlying IB-CH in the online phase. Besides that, Hash is a part of its verification algorithm. Thus, according to the computational costs of Hash and Col presented in Table 1, our optimized construction reduces the computational cost by $(1 - E/2E) \cdot 100\% = 50\%$ in the online phase and costs $3P + E_T + 2R - (P + E + 2E_T + R) \approx 2P$ less in the verification process compared with the generic OO-IBS construction [13]. In a word, ours improves the performance of both of the signing and verifying processes markedly.

7. CONCLUSION

In this paper, we propose an efficient identity-based chameleon hash (IB-CH) scheme for mobile devices, achieving ID-B-CollRes security in the standard model. Both of the theoretical and experimental analyses indicate that our IB-CH scheme is efficient, especially the experimental analysis showing that compared with the only existing IB-CH scheme in the standard model, our scheme reduces running time by approximately 73.5%, 55.0%, 51.4% and 53.1% in Setup, KeyGen, Hash and Col algorithms respectively, and reduces the size of public parameters by about 98.1% when the length of identity is 256 bits. We further applied our scheme to optimize the existing generic OO-IBS construction. Compared to the original one, our optimized construction improves the performance of the signing and verifying processes significantly.

8. REFERENCES

[1] Adi Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO'84*, 1984, pp. 47–53.

[2] Shidi Xu, Yi Mu, and Willy Susilo, "Online/offline signatures and multisignatures for AODV and DSR routing security," in *Proc. ACISP'06*, 2006, pp. 99–110.

[3] Adi Shamir and Yael Tauman, "Improved online/offline signature schemes," in *Proc. CRYPTO'01*, 2001, pp. 355–367.

[4] Hugo Krawczyk and Tal Rabin, "Chameleon signatures," in *Proc. NDSS'00*, 2000.

[5] Giuseppe Ateniese and Breno de Medeiros, "On the key exposure problem in chameleon hashes," in *Proc. SCN'04*, 2004, pp. 165–179.

[6] Giuseppe Ateniese, Bernardo Magri, Daniele Venturi, and Ewerton R. Andrade, "Redactable blockchain - or - rewriting history in bitcoin and friends," in *Proc. EuroS&P'17*, 2017, pp. 111–126.

[7] Mojtaba Khalili, Mohammad Dakhilalian, and Willy Susilo, "Efficient chameleon hash functions in the enhanced collision resistant model," *Inf. Sci.*, vol. 510, pp. 155–164, 2020.

[8] Jan Camenisch, David Derler, Stephan Krenn, Henrich C. Pöhls, Kai Samelin, and Daniel Slamanig, "Chameleon-hashes with ephemeral trapdoors - and applications to invisible sanitizable signatures," in *Proc. PKC'17*, 2017, pp. 152–182.

[9] David Derler, Kai Samelin, and Daniel Slamanig, "Bringing order to chaos: The case of collision-resistant chameleon-hashes," in *Proc. PKC'20*, 2020, pp. 462–492.

[10] Xiangyu Liu, Shengli Liu, and Dawu Gu, "Tightly secure chameleon hash functions in the multi-user setting and their applications," in *Proc. ACISP'20*, 2020, pp. 664–673.

[11] David Derler, Stephan Krenn, Kai Samelin, and Daniel Slamanig, "Fully collision-resistant chameleon-hashes from simpler and post-quantum assumptions," in *Proc. SCN'20*, 2020, pp. 427–447.

[12] Chunhui Wu, Lishan Ke, and Yusong Du, "Quantum resistant key-exposure free chameleon hash and applications in redactable blockchain," *Inf. Sci.*, vol. 548, pp. 438–449, 2021.

[13] Zhikang Xie, Qingni Shen, Cong Li, Jisheng Dong, and Yuejian Fang, "Identity-based chameleon hash without random oracles and application in the mobile internet," in *Proc. ICC'21*, 2021, pp. 1–6.

[14] Giuseppe Ateniese and Breno de Medeiros, "Identity-based chameleon hash and applications," in *Proc. FC'04*, 2004, pp. 164–180.

[15] Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo, "Id-based chameleon hashes from bilinear pairings," *IACR Cryptol. ePrint Arch.*, Report 2003/208, 2003.

[16] Xiaofeng Chen, Fangguo Zhang, Willy Susilo, Haibo Tian, Jin Li, and Kwangjo Kim, "Identity-based chameleon hash scheme without key exposure," in *Proc. ACISP'10*, 2010, pp. 200–215.

[17] Feng Bao, Robert H. Deng, Xuhua Ding, Junzuo Lai, and Yunlei Zhao, "Hierarchical identity-based chameleon hash and its applications," in *Proc. ACNS'11*, 2011, pp. 201–219.

[18] Chun-Hui Wu, Qin Li, and Chuan Lin, "New identity-based key-exposure free chameleon hash from bilinear pairings," *J. Networks*, vol. 9, no. 7, pp. 1756–1763, 2014.

[19] Xiaofeng Chen, Fangguo Zhang, Willy Susilo, Haibo Tian, Jin Li, and Kwangjo Kim, "Identity-based chameleon hashing and signatures without key exposure," *Inf. Sci.*, vol. 265, pp. 198–210, 2014.

[20] Peng Yi, Jiguo Li, Chengdong Liu, Jinguang Han, Huaqun Wang, Yichen Zhang, and Yu Chen, "An efficient identity-based signature scheme with provable security," *Inf. Sci.*, vol. 576, pp. 790–799, 2021.

[21] Fuchun Guo, Willy Susilo, and Yi Mu, "Digital signatures without random oracles," in *Introduction to Security Reduction*, pp. 173–192. Springer, 2018.

[22] Craig Gentry, "Practical identity-based encryption without random oracles," in *Proc. EUROCRYPT'06*, 2006, pp. 445–464.

[23] Dan Boneh and Xavier Boyen, "Short signatures without random oracles," in *Proc. EUROCRYPT'04*, 2004, pp. 56–73.

[24] Angelo De Caro and Vincenzo Iovino, "jPBC: Java pairing based cryptography," in *Proc. ISCC'11*, 2011, pp. 850–855.