IEICE TRANSACTIONS

on Information and Systems

VOL. E103-D NO. 2
FEBRUARY 2020

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

IEICE TRANS. INF. & SYST., VOL.E103–D, NO.2 FEBRUARY 2020

INVITED PAPER *Special Section on Security, Privacy, Anonymity and Trust in Cyberspace Computing and Communications*

White-Box Implementation of the Identity-Based Signature Scheme in the IEEE P1363 Standard for Public Key Cryptography

Yudi ZHANG[†,††], **Debiao HE**[†,††a)], **Xinyi HUANG**[†††,††††], **Ding WANG**[††,†††††], **Kim-Kwang Raymond CHOO**[††††††], ***Nonmembers***, ***and*** **Jing WANG**[†,††], ***Student Member***

SUMMARY Unlike black-box cryptography, an adversary in a white-box security model has full access to the implementation of the cryptographic algorithm. Thus, white-box implementation of cryptographic algorithms is more practical. Nevertheless, in recent years, there is no white-box implementation for public key cryptography. In this paper, we propose the first white-box implementation of the identity-based signature scheme in the IEEE P1363 standard. Our main idea is to hide the private key to multiple lookup tables, so that the private key cannot be leaked during the algorithm executed in the untrusted environment. We prove its security in both black-box and white-box models. We also evaluate the performance of our white-box implementations, in order to demonstrate utility for real-world applications.

key words: *white-box implementation, white-box security, IEEE P1363, identity-based signature, key extraction*

1. Introduction

White-box cryptography was first introduced by Chow et al. [1], [2] in 2002, and is designed to prevent software implementation of cryptographic algorithm from being attacked in untrusted envrionments. Specifically, the key purpose of white-box cryptography is to ensure the *confidentiality of secret keys*. Since the first white-box implementations of DES and AES algorithms [1], [2], a number of other white-box implementations have been proposed in the literature [3], [4].

In the trusted environment, an adversary knows the algorithm of the cryptographic system. The adversary can also require a number of inputs and obtain outputs from the program. However, the adversary does not have the permission to access the internal process of the program's execution. In practice, an adversary can also observe and modify the algorithm's implementation to obtain the internal details, such as the secret key. Many side-channel attacks have been proposed recently, most of them can be mounted on the existing cryptographic systems, such as timing, power, and fault analysis attacks. For example, the digital rights management (DRM) is commonly used to restrict the use of proprietary hardware and copyrighted works. In the example shown in Fig. 1, a broadcasting company wishes to distribute their digital content (e.g., music and movies) on the Internet, and set different permissions to the users such that only paying users can access the purchased content. However, these users should not be able to copy or re-distribute the content. Therefore, the provider should encrypt the content m first, i.e., computes $c = Enc_k(m)$, then distributes the entrypted content c on the public network. If the user has the license to access the content, then the Rights Expression Manager can parse the user's authentication and decrypt the encrypted content, i.e., the corresponding decryption program D computes $m = Dec_k(c)$ to obtain the original content. However, iTunes DRM has been reportedly cracked by Johansen [5], where the vulnerability can be exploited to re-distribute the content without authentication. Similar vulnerabilities in iOS DRM applications have been revealed by D'Orazio and Choo [6], which can be exploited to gain access to copyrighted materials for free.

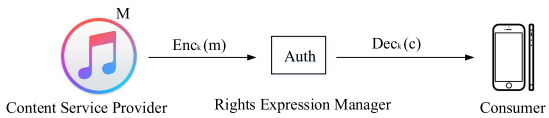


Fig. 1 A typical DRM architecture

As more DRM services are offered via mobile devices / applications (apps), it is vital to ensure the security of DRM and other services/apps, for example the use of cryptographic tools such as encryption and digital signature schemes. The latter is indispensable in the Internet especially in e-commerce, due to its capability to demonstrate the validity of user's message and identity. Formally, a valid digital signature ensures that the message was generated by a known signer, the signer cannot deny his/her signature, and that the integrity of the message has not been compromised. To avoid the limitations inherent in public key-based

Manuscript received March 3, 2019.
Manuscript revised June 19, 2019.
Manuscript publicized September 27, 2019.

[†]The authors are with Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China.
[††]The authors are with State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China.
[†††]The author is with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China.
[††††]The author is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou 350117, China.
[†††††]The author is with School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.
[††††††]The author is with Department of Information Systems and Cyber Security and the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA.
a) E-mail: hedebiao@163.com
DOI: 10.1587/transinf.2019INP0004

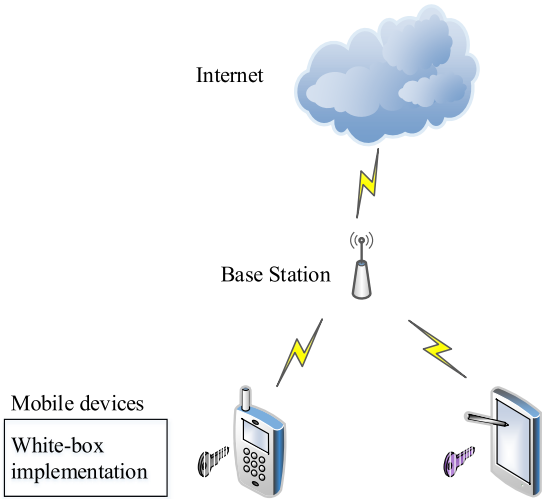


Fig. 2 A use case for white-box implementation in wireless environment

digital signature schemes, such as those of [7]–[9], Shamir introduced the first identity-based cryptography [10]. Since the seminal work of Shamir, many other identity-based signature (IBS) schemes, such as those of [11]–[14], have been proposed in the literature.

While identity-based digital signature is a topic that has been extensively studied, there is not any known white-box implementation of identity-based signature scheme. Thus, this is the focus and contribution of the work in this paper. Specifically, in our study, we focus on the white-box implementation of identity-based signature scheme in the IEEE P1363 standard for public key cryptography [15]. As far as we know, this is the first white-box implementation of identity-based signature scheme (in the IEEE P1363 standard). Our method is lightweight, and meets the white-box security requirement. As shown in Fig. 2, our method can be implemented in an untrusted wireless environment, including on mobile devices such as Android or iOS device. Specifically, the Sign algorithm is implemented on some user devices in the white-box model. Therefore, the malicious applications or hackers obtaining user's private key is impossible. Moreover, even if the device is lost, no one can get the user's private key.

In Sect. 2, we introduce related white-box cryptography literature, prior to presenting the notations, the identity-based signature scheme in the IEEE P1363 standard, mathematical assumptions and the definitions of white-box security in Sect. 3. In Sect. 4, we propose our white-box implementation of the identity-based signature scheme in the IEEE P1363 standard, and give the description of the detailed construction. In Sect. 5, we lay special stress on analyzing the black-box and white-box security. We implement our proposed method on a personal computer (PC), then we show and evaluate the implementation performance in Sect. 6. We point that our method is efficient and secure in the industrial area and the real-world applications. In the last section, we conclude the paper.

2. Related Work

White-box cryptography (WBC) is designed to protect software implementations of cryptographic algorithms when the software is running on untrusted environment, in the sense that the adversary has full access to the implementation [1], [2]. Chow *et al.* [1], [2] proposed the first white-box implementation for both DES and AES algorithms, and the authors also introduced the White-Box Attack Context (WBAC). In WBAC, the adversary seeks to extract the secret keys from the implementation. In recent years, many other white-box implementations have also been put forward, such as white-box implementations of DES and AES [3], [4], many of these schemes have been shown to be vulnerable to practical key extraction or table-decomposition attacks [16]–[18]. For example, using linear algebra, Lepoint *et al.* [18] demonstrated how Lepoint's construction can be broken. Biryukov *et al.* [19] also broke Chow *et al.*'s construction in 2014.

Billet *et al.* [16] proposed an effective cryptanalysis for white-box implementations of AES algorithm in 2004. They used algebraic cryptanalysis to analyze specific lookup tables, and removed the non-linear parts of the internal implementation. In a later work, Michiels *et al.* [20] proposed an improved cryptanalysis, which can be used to analyze a generic class of white-box implementations.

Biryukov *et al.* [19] also showed that the white-box implementations of AES and DES [1], [2] can be identified as a 3-layer ASA (affine-substitution-affine) structure, and they proposed a more secure structure (i.e., a 5-layer ASASA construction). Since the work of Biryukov *et al.* [19], other researchers [21], [22] have studied the decomposition of secret nonlinear and linear layers. Theoretically, the more layers that are employed, the more secure the construction is. However, Biryukov *et al.* [23] also showed that even a 9-layer construction SASASASAS is vulnerable.

Delerabl *et al.* [24] proposed a notion of *incompressibility*: an adversary has full access to the white-box implementation, but generate a program with the same function and dramatically small size is impossible. Such a notion is also referred to as *weak white-box* [19] or *space hardness* [25] in the literature. In this notion, the adversary cannot extract the key from the white-box implementation of the cryptographic algorithm, if the implementation is large and incompressible.

More recently in 2016, Bellare *et al.* [26] utilized a large encryption key to protect the key, which is called the bounded-retrieval model (BRM), and Fouque *et al.* [27] proposed the first construction with provable security guarantee. They also introduced a new definition of incompressibility (i.e. weak and Fouque *et al.* incompressibility).

A number of differential cryptanalysis techniques can be used to crack white-box implementations [1], [28], especially on white-box implementations for DES. For example, Chow *et al.* [1] showed that their white-box implementations of DES is vulnerable. They then introduced an attack sim-

ilar to differential power analysis, i.e. statistical bucketing attack. The statistical bucketing attack method has been improved subsequently by Link and Neumann [28].

While there are numerous identity-based signature schemes, they are generally not white-box attack resilience, and we are not aware of any white-box implementation for identity-based signature scheme. Hence, our research has filled the gap in white-box cryptography.

3. Preliminaries

We let S denote a set or distribution, and $a \xleftarrow{r} S$ denote that a is randomly selected from S. In this paper, n denotes the security parameter, for any polynomial p, if the equation $\mu(n) = O(1/p(n))$ holds, then the function $\mu(n)$ is negligible. The trusted key generation center denotes in KGC, the probabilistic polynomial time algorithm denotes in P.P.T. H_1 and H_2 are two secure hash functions, such that $H_1 : \{0,1\}^* \to \mathbb{Z}_p$, and $H_2 : \{0,1\}^* \times \mathbb{G}_3 \to \mathbb{Z}_p$.

Let Setup be the algorithm that, given the security parameter n, it outputs the bilinear map parameters $(g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, e)$. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic additive groups, g_1, g_2 are generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively, $\mathbb{G}_3$ is a multiplicative group, there exists an efficient bilinear map such that $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$, which contains properties as follows:

1. For all $x_1, x_2 \in \mathbb{G}_1$ and $y_1, y_2 \in \mathbb{G}_2$, $e(x_1 + x_2, y_1) = e(x_1, y_1)e(x_2, y_1)$ and $e(x_1, y_1 + y_2) = e(x_1, y_1)e(x_1, y_2)$.
2. For all $0 \neq x \in \mathbb{G}_1$, there exists $y \in \mathbb{G}_2$ such that $e(x, y) \neq 1$.
3. For all $0 \neq x \in \mathbb{G}_2$, there exists $y \in \mathbb{G}_1$ such that $e(x, y) \neq 1$.
4. There exists an efficient isomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$, such that $g_1 = \phi(g_2)$.

3.1 The Identity-Based Signature Scheme in IEEE P1363 Standard

In this section, we review the identity-based signature in the IEEE P1363 standard [13] briefly. The detailed algorithms are described below:

1. **Setup:** Taken as input the security parameter n, the KGC outputs the system parameters params as follows:
   a. Chooses $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3$ and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$.
   b. Picks a random generator Q_2 of $\mathbb{G}_2$, and calculates $Q_1 = \phi(Q_2) \in \mathbb{G}_1$.
   c. Randomly selects $s \xleftarrow{r} \mathbb{Z}_p$, sets s as the master secret key, then calculates $R = sQ_2$ and $g = e(Q_1, Q_2)$.
   d. Sets and outputs the system parameters params $= (R, g, Q_1, Q_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, e)$ available.
2. **Extract:** Taken as input a user's identity ID, the KGC outputs the user's private key as follows:
   a. Computes the identity element $h_{ID} = H_1(ID)$ where $h_{ID} \in \mathbb{Z}_p$.
   b. Outputs $K_{ID} = (h_{ID} + s)^{-1} Q_1$.
3. **Sign:** Taken as input a message m, the user's identity ID, the signer outputs the signature σ as follows:
   a. Randomly selects $r \xleftarrow{r} \mathbb{Z}_p$, computes $u = g^r$.
   b. Computes $h = H_2(m, u)$ and $S = (r + h)K_{ID}$.
   c. Outputs the signature $\sigma = (h, S)$.
4. **Verify:** Taken as input the signature σ, the corresponding message m, and the identity ID, this algorithm should check the validation of the signature. The verifier executes the steps as follows:
   a. Computes $h_{ID} = H_1(ID)$.
   b. Computes $u = \frac{e(S, h_{ID}Q_2 + R)}{e(Q_1, Q_2)^h}$.
   c. If $h = H_2(m, u)$, then outputs 1; otherwise, outputs 0.

3.2 Mathematical Assumptions

Definition 1. *We assume that there exists bilinear map groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_3$, P is the generator of $\mathbb{G}_1$, Q is the generator of $\mathbb{G}_2$. The q-Strong Diffie-Hellman problem (q-SDHP) in $(\mathbb{G}_1, \mathbb{G}_2)$ is described as follows: taken as input $(q+2)$-tuple $(P, Q, xQ, x^2Q, \ldots, x^qQ)$, and output that $(c, \frac{1}{c+x}P)$ where $c \in \mathbb{Z}_p^*$. A P.P.T algorithm $\mathcal{A}$ solves q-SDHP in $(\mathbb{G}_1, \mathbb{G}_2)$ with the advantage ϵ if*

$$\Pr[\mathcal{A}(P, Q, xQ, x^2Q, \ldots, x^qQ) = (c, \frac{1}{c+x}P)] \geq \epsilon.$$

We say that q-SDHP in $(\mathbb{G}_1, \mathbb{G}_2)$ is infeasible if all P.P.T algorithms can solve q-SDHP in $(\mathbb{G}_1, \mathbb{G}_2)$ with a negligible advantage ϵ.

Definition 2. *Let $\mathbb{G}$ be a cyclic group of prime order q. The DL problem in $\mathbb{G}$ is to compute $a \in \mathbb{Z}_q$ for given (P, Y) where $Y = aP \in \mathbb{G}$. A P.P.T algorithm $\mathcal{A}$ solves DL problem in $\mathbb{G}$ with the advantage ϵ if*

$$\Pr[\mathcal{A}(P, Y) = a : a \in \mathbb{Z}_p, Y = aP] \geq \epsilon.$$

We say that the DL problem in $\mathbb{G}$ is infeasible if all P.P.T algorithm can solve the DL problem in $\mathbb{G}$ with a negligible advantage ϵ.

3.3 White-Box Security

We adapt existing definitions of white-box security [19], [25], presented below.

Definition 3. *White-Box Attack Context (WBAC) [2]:*

- *an attack software has full privileges and shares a host with the cryptographic software, and it has full access to the implementation of the algorithm;*

- *cryptographic software can be executed dynamically and observed (i.e. the instantiated cryptographic keys);*
- *the internal details of the algorithm are completely visible and alterable.*

Definition 4. *Strong White-Box Security: We assume that the pair of algorithm (E, D) is a symmetric key scheme, and K is the secret key. Let O_{E_K} be a function that computes E_K. Given full access to O_{E_K}, if obtain a function $\mathcal{D}'$ equivalent to D_K is computationally hard, then we say that O_{E_K} is a secure strong white-box implementation for E_K.*

Based on the definition in [19] and our proposed white-box implementation of identity-based signature scheme, we give the definition of weak white-box security.

Definition 5. *Weak White-Box Security: Let the pair of algorithms (S, V) be a signature scheme, and K is the private key. We generate an equivalent key set $\mathfrak{F}(K)$ for K, and it is trivial to obtain an algorithm from $\mathfrak{F}(K)$ which is equivalent to S_K. The function O_{S_K} is a T-secure weak white-box implementation for S_K, if given the full access to O_{S_K}, to obtain the K of length less than T from $\mathfrak{F}(K)$ is computationally hard.*

That is, when an adversary is given the full access to the secure weak white-box implementation to find out any compact equivalent function smaller than T, it is computationally hard.

4. White-Box Implementation of the Identity-Based Signature Scheme in IEEE P1363

In this section, we propose our white-box implementation of the identity-based signature scheme in the IEEE P1363 standard.

Our proposed method consists of the following five algorithms, namely: Setup, Extract, WhiteBoxKeyGen, Sign and Verify.

1. **Setup:** Taken as input the security parameter n, the KGC outputs the system parameters params as follows:
   a. Chooses $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3$ and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$.
   b. Picks a random generator Q_2 of $\mathbb{G}_2$, and calculates $Q_1 = \phi(Q_2) \in \mathbb{G}_1$.
   c. Randomly selects $s \in \mathbb{Z}_p$, sets s as the master secret key, and calculates $R = sQ_2$ and $g = e(Q_1, Q_2)$.
   d. Sets params $= (R, g, Q_1, Q_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, e)$.
2. **Extract:** Taken as input a user's identity ID, the KGC outputs the user's private key as follows:
   a. Computes the identity element $h_{ID} = H_1(ID)$ where $h_{ID} \in \mathbb{Z}_p$.
   b. Outputs $K_{ID} = (h_{ID} + s)^{-1} Q_1$.
3. **WhiteBoxKeyGen:** Taken as input the user with the identity ID and params, the KGC outputs white-box keys as follows:
   a. Randomly selects $x_1, x_2 \cdots, x_k \overset{r}{\leftarrow} \mathbb{Z}_p$ where k is a number greater than or equal to 256.
   b. Computes $\{u_1 = g^{x_1}, u_2 = g^{x_2}, \cdots, u_k = g^{x_k}\}$ and $\{X_1 = x_1 K_{ID}, X_2 = x_2 K_{ID}, \cdots, X_k = x_k K_{ID}\}$.
   c. Randomly selects $y_1, y_2 \cdots, y_{256} \overset{r}{\leftarrow} \mathbb{Z}_p$.
   d. Computes $\{v_1 = g^{y_1}, v_2 = g^{y_2}, \cdots, v_{256} = g^{y_{256}}\}$ and $\{Y_1 = K_{ID} + y_1 K_{ID}, Y_2 = 2K_{ID} + y_2 K_{ID}, \cdots, Y_{256} = 2^{255} K_{ID} + y_{256} K_{ID}\}$.
   e. Deletes $\{x_1, x_2 \cdots, x_k\}$ and $\{y_1, y_2 \cdots, y_{256}\}$.
   f. Sends u_i, X_i and Y_i to the signer, and makes v_i public.
4. **Sign:** Taken as input a message m, the user's identity ID, the signer outputs the signature σ as follows:
   a. Generates a k bits number r randomly, where r is a binary and represented as $r_k \cdots r_2 r_1$. Computes $u' = \prod_{i:r_i=1} u_i$, and $S_1 = \sum_{i:r_i=1} X_i$.
   b. Computes $h = H_2(m, u')$, where h is a binary and represented as $h_{256} \cdots h_2 h_1$.
   c. Computes $S_2 = \sum_{i:h_i=1} Y_i$, and sets $S' = S_1 + S_2$.
   d. Outputs $\sigma = (h, S')$.
5. **Verify:** Taken as input the signature σ, the corresponding message m, and the identity ID, this algorithm should check the validation of the signature. The verifier executes the steps as follows:
   a. Binary h is represented as $h_{256} \cdots h_2 h_1$, and computes $t_1 = \prod_{i:h_i=1} v_i$.
   b. Computes $t_2 = \frac{e(S', h_{ID} Q_2 + R)}{e(Q_1, Q_2)^h}$, and sets $u = \frac{t_2}{t_1}$.
   c. Computes $h' = H_2(m, u)$. If $h = h'$, then outputs 1; otherwise, outputs 0.

5. Security Analysis

We show the black-box and white-box security analysis separately in this section.

5.1 Black-Box Security Analysis

First, we prove that our proposed method achieves the security requirement in the black-box model. According to existing security model [12], [13], [29] for the identity-based signatures, an identity-based signature scheme is existentially unforgeable under adaptive chosen-message attacks.

Definition 6. *If an IBS scheme is existentially unforgeable under adaptive chosen message and identity attacks, then for any P.P.T adversary $\mathcal{A}$ who interacts with a challenger C will play the game as follows:*

1. *C executes Setup algorithm to produce the system parameters, then returns it to $\mathcal{A}$.*
2. *$\mathcal{A}$ performs the two queries as follows:*

a. *Query on* Extract *oracle. On input an identity ID, C outputs a private key which corresponds to the identity ID.*
b. *Query on* Sign *oracle. On input an identity ID and a message m, C outputs a signature which corresponds to the ID's private key.*

3. *$\mathcal{A}$ outputs the tuple (ID^*, m^*, σ^*), where such ID^* have never been queried to* Extract *oracle, and (ID^*, m^*) have never been queried to* Sign *oracle. If* Verify *accepts (ID^*, m^*, σ^*), then $\mathcal{A}$ wins the game.*

$\mathcal{A}$ can win this game with a negligible advantage.

Lemma 1. *[13] Given an adaptively chosen message and the identity to the attacker $\mathcal{A}$, $\mathcal{A}$ can make q_{h_i} queries to the oracle H_1 and the oracle H_2, q_s queries to the signing oracle. Within the time bound t, if $\mathcal{A}$ can produce a forgery with the advantage $\epsilon \geq 10(q_s+1)(q_s+q_{h_2})/2^n$, then there exists another algorithm $\mathcal{B}$ which can solve the q-SDHP problem with the advantage $t' \leq 120686_{q_{h_2}} t/\epsilon$.*

Proof. In order to apply the forking lemma [30], for the P.P.T algorithm $\mathcal{B}$, on input $(P, Q, xQ, x^2Q, \ldots, x^qQ)$, it finds a pair $(c, \frac{1}{c+x}P)$. Similar to the proof in [13], $\mathcal{B}$ computes $\sum_{i=0}^{q-2} d_i\psi(x^iQ) = \frac{1}{x+\omega_i}G$, where G is the generator of $\mathbb{G}_1$.

Firstly, $\mathcal{B}$ initializes l a counter, and sets $l = 1$, then executes $\mathcal{A}$ on the input (H_{pub}, ID^*), where $H_{pub} \in \mathbb{G}_2$ is the public key.

- H_1-queries: On input an identity $ID \in \{0,1\}$, if $ID = ID^*$, then $\mathcal{B}$ selects $w^* \xleftarrow{r} \mathbb{Z}_p^*$ randomly and returns it. Otherwise, $\mathcal{B}$ selects $w_l \xleftarrow{r} \mathbb{Z}_p^*$ randomly, sets $w = w_l$, and answers w together with the increments l. Then, $\mathcal{B}$ stores (ID, w) in the list L_1.
- Key extraction queries: For an input $ID \neq ID^*$, $\mathcal{B}$ searches the list L_1 and recovers the pair (ID, w), then computes $(1/(x+w))G$ and returns it.
- Sign queries: For an input tuple (m, ID), $\mathcal{B}$ randomly selects $S \xleftarrow{r} \mathbb{G}_1$, $h \xleftarrow{r} \mathbb{Z}_p^*$, and computes $u = e(S, H_1(ID)H + H_{pub})e(G,H)^{-h}$, then sets $H_2(m,u) = h$, if $H_2(m,u)$ is already set, then $\mathcal{B}$ aborts.

If the adversary $\mathcal{A}$ has no knowledge of the private key, but he still can simulate the tuple (u, h, S), then there exists a P.P.T algorithm $\mathcal{A}'$ which can employ $\mathcal{A}$ to generate two signatures (m, u, h_1, S_1) and (m, u, h_2, S_2) where $h_1 \neq h_2$ with the time $t' \leq 120686_{q_{h_2}} t/\epsilon$. Both signatures can pass the Verify algorithm.

Algorithm $\mathcal{B}$ executes $\mathcal{A}'$ and to generate two different forgeries (m^*, u, h_1, S_1) and (m^*, u, h_2, S_2), the two messages m^*, u are the same. $\mathcal{B}$ searches the list L_1 and gets the pair (ID^*, w^*). Note that, $w^* \notin \{w_1, \ldots, w_{q-1}\}$, and the probability is at least $1 - q/2^n$. If the two forgeries can pass the Verify algorithm, then we have

$$e((h_1 - h_2)^{-1}(S_1 - S_2), (w^* + x)H) = e(G, H),$$

due to $(h_1 - h_2)^{-1}(S_1 - S_2) = \frac{1}{w^*+x}G$, then $\mathcal{B}$ can extract $\sigma^* = \frac{1}{w^*+x}G$.

Therefore, if $\mathcal{A}$ can forge a signature with the advantage $\epsilon \geq 10(q_s+1)(q_s+q_{h_2})/2^n$ in time t, then, $\mathcal{B}$ can solve q-SDHP within time t'. □

Table 1 Lookup table for u_i

Index	
1	g^{x_1}
2	g^{x_2}
⋯	⋯
i	g^{x_i}
⋯	⋯
k	g^{x_k}

Table 2 Lookup table for X_i

Index	
1	x_1K_{ID}
2	x_2K_{ID}
⋯	⋯
i	x_iK_{ID}
⋯	⋯
k	x_kK_{ID}

Table 3 Lookup table for Y_i

Index	
1	$K_{ID} + y_1K_{ID}$
2	$2K_{ID} + y_2K_{ID}$
⋯	⋯
i	$2^{i-1}K_{ID} + y_iK_{ID}$
⋯	⋯
256	$2^{255}K_{ID} + y_{256}K_{ID}$

Table 4 Lookup table for v_i

Index	
1	g^{y_1}
2	g^{y_2}
⋯	⋯
i	g^{y_i}
⋯	⋯
256	$g^{y_{256}}$

5.2 White-Box Security Analysis

We proved that our proposed method is existentially unforgeable under chosen-message attacks in the previous subsection. Now, we analyze the white-box security of our white-box implementation of identity-based signature in the IEEE P1363 standard.

Lemma 2. *If a P.P.T algorithm can compute and obtain the private key K_{ID} from the public parameter, then it can solve the DL problem.*

Proof. In our proposed method, the public parameter includes both params and white-box key – see Table 1, Table 2, Table 3 and Table 4.

In the **WhiteBoxKeyGen** phase, KGC deletes $\{x_1, x_2, \ldots, x_k\}$ and $\{y_1, y_2, \ldots, y_{256}\}$. If a P.P.T adversary $\mathcal{A}$ can

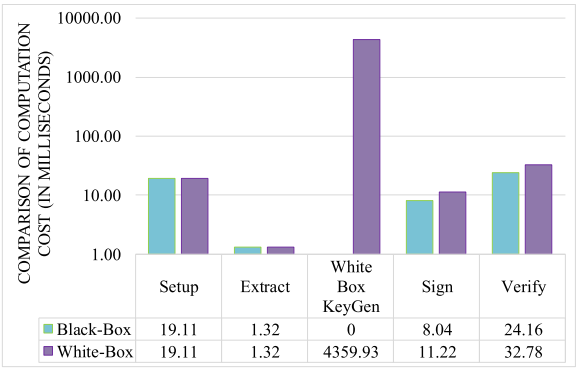


Fig. 3 Computation cost (in milliseconds): comparative summary

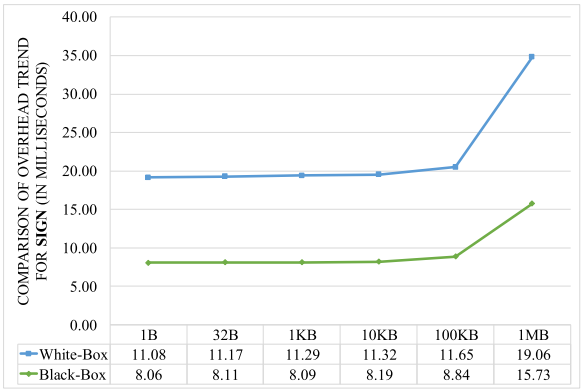


Fig. 4 Time costs for messages of different sizes in the sign algorithm (in milliseconds)

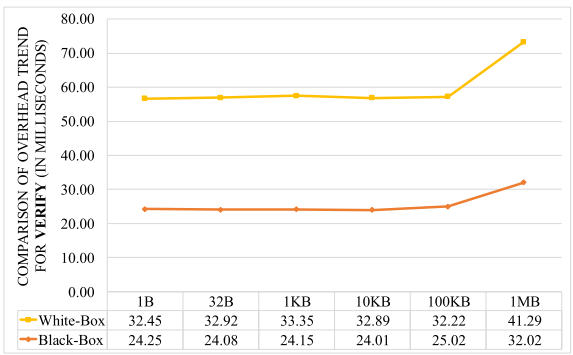


Fig. 5 Time costs for messages of different sizes in the verify algorithm (in milliseconds)

compute K_{ID} from Table 2 and Table 3, then it means that there exists a P.P.T algorithm $\mathcal{A}'$ which can solve the DL problem in a non-negligible advantage.

In other words, our proposed method meets the requirement of weak white-box security. □

Definition 7. *White-Box Diversity [2]: If we encode the scheme implementation steps, and can count the possible encoded steps, then this is the white-box diversity. The greater the diversity value, the safer the scheme.*

In our white-box implementation of the identity-based signature scheme in IEEE P1363 standard, the diversity is $2^{n+log_2 k}$.

6. Performance Evaluation

In this section, we implement our proposed method using MIRACL Cryptographic SDK [31], then we show and evaluate the implementation performance. In addition, we compare our proposed method with the original IEEE P1363 signature scheme. The implementation of our method is deployed on a PC (with an Intel Xeon E3-1230 v5 processor, 12GB memory and the Microsoft Windows 10 operating system). The curve we used to evaluate is BN curve which achieves the AES-128 security.

The comparative summary between our method and the original IEEE P1363 scheme is presented in Fig. 3, where the black-box denotes the original scheme and the white-box is our method. Note that only WhiteBoxKeyGen algorithm is employed in our method, and it is executed by the KGC. Setup and Extract algorithms are same for the both schemes; thus, they are omitted from the comparative summary. The time costs for both Sign and Verify algorithms in the proposed and original schemes are similar, with the exception of the time costs for the WhiteBoxKeyGen algorithm. However, WhiteBoxKeyGen is executed by the KGC, so it has little effect on the user.

We also evaluate using messages of different lengths in both Sign and Verify algorithms. As shown in Fig. 4, the lengths of the messages used are 1byte, 32bytes, 1K bytes, 10K bytes, 100K bytes and 1M bytes. With the exception of the message of 1M-byte in length, the messages are signed for approximately 11 ms and 8 ms in white-box and black-box implementation, respectively. It takes about 19 ms in white-box implementation and 15 ms in black-box implementation, respectively, when the length of the message is 1M bytes.

Similarly, shown in Fig. 5 is the time costs for the Verify algorithm. For messages less than or equal to 100K bytes, it takes almost 32 ms and 24 ms in white-box and black-box implementation, respectively. However, when the message reaches 1M bytes, the time costs for white-box and black-box implementations are respectively 41 ms and 32 ms.

7. Conclusion

White-box cryptanalysis and attacks are more crucial than black-box security in a real-world application, particularly in terms of ensuring the security of a user secret key.

In this paper, we proposed a novel white-box implementation for the identity-based signature scheme in the IEEE P1363 standard which is efficient and secure. Specifically, this allows us to produce a valid signature in a white-box model without leaking the private key. The security analysis demonstrated that our method can meet the white-box security requirement. According to the performance evaluation, our proposed method showed that it is potentially useful in the industrial area and the real world appli-

cations.

In the future, we intend to conduct a more comprehensive evaluation, for example using popular consumer devices (e.g., a broad range of Android, iOS, Windows Phones devices, as well as IoT devices).

Acknowledgments

We greatly appreciate the invaluable suggestions provided by the anonymous reviewers and the associate editor. The work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802500, and in part by the National Natural Science Foundation of China under Grant 61572379, Grant 61501333, Grant 61822202, Grant 61872089, Grant 61902070 and Grant 61972094.

References

[1] S. Chow, P. Eisen, H. Johnson, and P.C. Van Oorschot, "A white-box des implementation for drm applications," ACM Workshop on Digital Rights Management, vol.2696, pp.1–15, Springer, 2002.

[2] S. Chow, P. Eisen, H. Johnson, and P.C. Van Oorschot, "White-box cryptography and an aes implementation," International Workshop on Selected Areas in Cryptography, vol.2595, pp.250–270, Springer, 2002.

[3] J. Bringer, H. Chabanne, and E. Dottax, "White box cryptography: Another attempt," IACR Cryptology ePrint Archive, vol.2006, no.2006, p.468, 2006.

[4] M. Karroumi, "Protecting white-box aes with dual ciphers," International Conference on Information Security and Cryptology, vol.6829, pp.278–291, Springer, 2010.

[5] A. Orlowski, "iTunes DRM cracked wide open for GNU/Linux. seriously." https://www.theregister.co.uk/2004/01/05/itunes_drm_cracked_wide_open/. Jan 5, 2004.

[6] C. D'Orazio and K.-K.R. Choo, "An adversary model to evaluate drm protection of video contents on ios devices," Computers & Security, vol.56, pp.94–110, 2016.

[7] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE transactions on information theory, vol.31, no.4, pp.469–472, 1985.

[8] M. Abdalla and L. Reyzin, "A new forward-secure digital signature scheme," International Conference on the Theory and Application of Cryptology and Information Security, vol.1976, pp.116–129, Springer, 2000.

[9] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," International journal of information security, vol.1, no.1, pp.36–63, 2001.

[10] A. Shamir, "Identity-based cryptosystems and signature schemes," Workshop on the theory and application of cryptographic techniques, vol.196, pp.47–53, Springer, 1984.

[11] F. Hess, "Efficient identity based signature schemes based on pairings," International Workshop on Selected Areas in Cryptography, vol.2595, pp.310–324, Springer, 2002.

[12] J.C. Choon and J.H. Cheon, "An identity-based signature from gap diffie-hellman groups," International workshop on public key cryptography, vol.2567, pp.18–30, Springer, 2003.

[13] P.S.L.M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," International conference on the theory and application of cryptology and information security, vol.3788, pp.515–532, Springer, 2005.

[14] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating eves algorithm and its application in fair electronic transactions in public clouds," IEEE Systems Journal, vol.13, no.2, pp.1478–1486, June 2019.

[15] IEEE Standards Association, "IEEE 1363-2000 - IEEE standard specifications for public-key cryptography," https://standards.ieee.org/standard/1363-2000.html, 2000.

[16] O. Billet, H. Gilbert, and C. Ech-Chatbi, "Cryptanalysis of a white box aes implementation," International Workshop on Selected Areas in Cryptography, vol.3357, pp.227–240, Springer, 2004.

[17] B. Wyseur, W. Michiels, P. Gorissen, and B. Preneel, "Cryptanalysis of white-box des implementations with arbitrary external encodings," International Workshop on Selected Areas in Cryptography, vol.4876 pp.264–277, Springer, 2007.

[18] T. Lepoint, M. Rivain, Y. De Mulder, P. Roelse, and B. Preneel, "Two attacks on a white-box aes implementation," International Conference on Selected Areas in Cryptography, vol.8282, pp.265–285, Springer, 2013.

[19] A. Biryukov, C. Bouillaguet, and D. Khovratovich, "Cryptographic schemes based on the asasa structure: Black-box, white-box, and public-key (Extended Abstract)," International Conference on the Theory and Application of Cryptology and Information Security, pp.63–84, Springer, 2014.

[20] W. Michiels, P. Gorissen, and H.D.L. Hollmann, "Cryptanalysis of a generic class of white-box implementations," International Workshop on Selected Areas in Cryptography, vol.5381, pp.414–428, Springer, 2008.

[21] A. Biryukov and A. Shamir, "Structural cryptanalysis of sasas," Journal of cryptology, vol.23, no.4, pp.505–518, 2010.

[22] J. Borghoff, L.R. Knudsen, G. Leander, and S.S. Thomsen, "Slender-set differential cryptanalysis," Journal of cryptology, vol.26, no.1, pp.11–38, 2013.

[23] A. Biryukov and D. Khovratovich, "Decomposition attack on SASASASAS," IACR Cryptology ePrint Archive, p.646, 2015.

[24] C. Delerablée, T. Lepoint, P. Paillier, and M. Rivain, "White-box security notions for symmetric encryption schemes," International Conference on Selected Areas in Cryptography, vol.8282, pp.247–264, Springer, 2013.

[25] A. Bogdanov and T. Isobe, "White-box cryptography revisited: Space-hard ciphers," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp.1058–1069, ACM, 2015.

[26] M. Bellare, D. Kane, and P. Rogaway, "Big-key symmetric encryption: Resisting key exfiltration," Annual Cryptology Conference, vol.9814, pp.373–402, Springer, 2016.

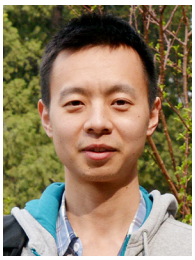
[27] P.-A. Fouque, P. Karpman, P. Kirchner, and B. Minaud, "Efficient and provable white-box primitives," International Conference on the Theory and Application of Cryptology and Information Security, vol.10031, pp.159–188, Springer, 2016.

[28] H.E. Link and W.D. Neumann, "Clarifying obfuscation: improving the security of white-box des," International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II, pp.679–684, IEEE, 2005.

[29] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," Journal of Cryptology, vol.22, no.1, pp.1–61, 2009.

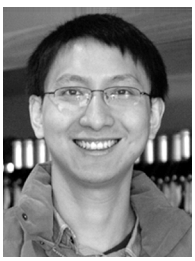
[30] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," Journal of cryptology, vol.13, no.3, pp.361–396, 2000.

[31] Miracl, "Miracl library," https://www.miracl.com/, 2017.



Yudi Zhang received the master's degree from Hubei University of Technology of China, in 2017. He is currently working toward the PhD degree in the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His main research interests include cryptography and information security, in particular, cryptographic protocols.

Debiao He received the PhD degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2009. He is currently a professor of the School of Cyber Science and Engineering, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.



Xinyi Huang received his Ph.D. degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia, in 2009. He is currently a Professor at the College of Mathematics and Informatics, Fujian Normal University, China. His research interests include cryptography and information security. He has published over 160 research papers in refereed international conferences and journals.

Ding Wang received the Ph.D. degree in information security from Peking University in 2017. He is currently supported by the Boya Post-Doctoral Fellowship in Peking University, China. He has authored over 40 papers at venues like ACM CCS and IEEE TDSC, and his papers get over 800 citations. His research interests mainly focus on password-based authentication and provable security. He received the Top-10 Distinguished Graduate Academic Award from Harbin Engineering University in 2013 and Peking University in 2016.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA), and has a courtesy appointment at the University of South Australia. In 2016, he was named the Cybersecurity Educator of the Year - APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also a Fellow of the Australian Computer Society, and an IEEE Senior Member.

Jing Wang received the B.S. degrees in computer science from from Wuhan University, Wuhan, China, in 2016. She is currently pursing a Ph.D degree in the School of Computer Science, Wuhan University, China. Her main research interests include cryptography and information security, in particular, secure cloud storage and cryptographic protocols.