Wildcarded Identity-Based Encryption with Constant-size Ciphertext and Secret Key

Dung Hoang Duong[1], Willy Susilo[1], and Viet Cuong Trinh[2*]
[1]*Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong, NSW 2500, Australia*
{hduong, wsusilo}@uow.edu.au
[2]*Faculty of Information and Communication Technology, Hong Duc University, 565 Quang Trung, Thanh Hoa, Viet Nam*
trinhvietcuong@hdu.edu.vn

Received: February 26, 2020; Accepted: June 5, 2020; Published: June 30, 2020

Abstract

Wildcard identity-based encryption (WIBE) is essentially a generalization of Hierarchical identity based encryption (HIBE) where at the time of encryption, the sender can decide to make the ciphertext decryptable by a whole range of users whose identities match a certain pattern, defined as a sequence of identities and wildcards. Almost existing WIBE schemes have a weakness, that is the large ciphertext size. Only very recently, at ESORICS'18, Kim et al. proposed the first WIBE scheme with constant size ciphertext (including four elements), however, the user's secret key size in their scheme is still large. In this paper, we propose a new WIBE scheme which is an improvement of Kim et al.'s scheme in term of all the ciphertext size, the secret key size and the decryption time. More precisely, in our scheme the ciphertext just contains three elements and user needs to keep only one element secret, all other elements in the user's secret key can be made public. For decrypting, user in our scheme only needs to compute two Pairings.

Keywords: wildcard identity based encryption, constant size ciphertex, constant size secret key, fast decryption.

1 Introduction

An Identity Base Encryption (IBE) scheme, introduced by Shamir [1] in 1984, is a public-key cryptosystem where any string is a valid public key such as email addresses and dates. It was until 2001 that Boneh and Franklin [2, 3] introduced the first practical IBE scheme using bilinear maps. In an IBE, the trusted third party, called a Private Key Generator (PKG), who has the master secret key, can issue a private key for each user with identity in the system, and hence IBE helps to eliminate the need for a public key distribution infrastructure. In a large network, the work of PKG would be burdensome in both private key generation, which will be computationally expensive, and identity verification as well as secure private key transmissions. Gentry and Silverberg [4] then came up with the idea of hierarchy identity-based encryption (HIBE) schemes which allows a root PKG to distribute the workload by delegating private key generation and identity authentication to lower-level PKGs. In an HIBE scheme [4, 5], root PKG needs to generate the private keys for only the identities of the first level, who in turn can use their private keys

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 11(2):74-86, June 2020
DOI:*10.22667/JOWUA.2020.06.30.074*

*Corresponding Author: Faculty of Information and Communication Technology, Hong Duc University, 565 Quang Trung, Thanh Hoa, Viet Nam, Tel: +84-948-921-740

to generate the private keys for identities of next level. Since then, several efficient HIBE schemes have been constructed [6, 7, 8].

One of the main application areas proposed for IBE is that of email encryption, in which given an email address, one can encrypt a message to the owner of the email address. Motivated by the fact that many email addresses correspond to groups of users rather than single individuals, Abdalla et al. [9, 10] introduced the notion of *wildcarded identity-based encryption* (WIBE), a generalisation of HIBE, which allows the encrypter to replace any component of the receipient identity with a *wildcard* so that any identity matching the *pattern* can decrypt. For instance, by encrypting a message to the pattern `*@cs.univ.edu`, then any user from the computer science department of `univ` would be able to decrypt it. The authors [9] then proposed three WIBE constructions from HIBE schemes in [6, 7, 8]. However, their ciphertext size grows linearly in the depth of the pattern. Abdalla et al. [9] also introduced a generic construction of a WIBE scheme from an HIBE scheme whose ciphertext size is as the same as that of HIBE scheme. Hence one can easily construct a constant size ciphertext WIBE scheme from any constant size ciphertext HIBE scheme (e.g., [7]). However, the secret key size grows exponentially with the depth of the pattern, this scheme is obviously impractical.

At ESORICS 2018, Kim et al. [11] reversed the method of Abdalla et al. [9] and proposed the first practical WIBE scheme, based on the HIBE scheme in [7], with constant size ciphertext; every ciphertext contains only four group elements. Their method is to let each user store an extra data for each non-wildcard identity in order to replace the identity by a wildcard. A pattern with l specific identity strings leads to a secret key with l additional elements and hence reduces the extra data values for each wildcard in the ciphertext as a single element; see [11] for more detail.

However, the drawback of all aforementioned schemes is that their secret key size is large, increasing linearly in the depth of the pattern; see Table 1 for more detail.

Our contributions. In this paper, we propose a new improved WIBE scheme which has following properties:

- constant-size of ciphertext: the ciphertext just includes three elements;
- constant-size of secret key: in our scheme, user just needs to keep only one element secret, all other elements can be made public. The secret key size of all current schemes grows linearly with the depth of pattern;
- fast decryption: to decrypt, user just needs to compute two Pairings in the prime order setting and L multiplicative operations, L is the pattern depth.

To the best of our knowledge, there exists only one WIBE scheme which achieves the constant-size of ciphertext property, that is the scheme at ESORICS'18 [11]. Compare to this scheme, our scheme is an improvement in term of all ciphertext-size (three elements compare to four elements), secret key size (one element compares to $O(L)$ elements) and decryption time (two pairing operations compare to three pairing operations plus L exponential operations). Our scheme and their scheme also achieve the same level of security, that is the CPA-selective security under a similar assumption. The weaknesses of our scheme compare to the scheme at ESORICS'18 are that, in our scheme the public storage is larger and our scheme is not scalable (the maximum number of identities is fixed at the setup phase). In addition, we note that in [11], the authors also show how to improve the security of their scheme with the trade-off on the efficiency.

2 Preliminaries

2.1 Identity-based Encryption

An IBE scheme is a tuple of four algorithms $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Encrypt}, \mathsf{Decrypt})$ providing the following functionality. The trusted authority runs Setup to generate a master key pair $(\mathsf{mpk}, \mathsf{msk})$. It publishes the master public key mpk and keeps the master secret key msk private. For an ID in the system, the trusted authority generates a secret key $d_{\mathsf{ID}} \leftarrow \mathsf{KeyDer}(\mathsf{msk}, \mathsf{ID})$ and sends it to the user with identity ID. To encrypt a message m to the user with identity ID, the sender computes the ciphertext $C \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{ID}, m)$, which can be decrypted by the user as $m \leftarrow \mathsf{Decrypt}(d_{\mathsf{ID}}, C)$. We refer to [3] for the security definitions for IBE schemes.

2.2 Hierarchical Identity-based Encryption

In a hierarchical identity-based encryption (HIBE) scheme, users are arranged in a tree of depth L with root being the trusted authority. The identity of a user at level $0 \leq l \leq L$ in the tree is given by a vector $\mathsf{ID} = (\mathsf{ID}_1, \cdots, \mathsf{ID}_l) \in (\{0,1\}^*)^l$. A HIBE scheme is a tuple of algorithms $\mathsf{HIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Encrypt}, \mathsf{Decrypt})$ providing the same functionality as an IBE scheme, except that a user $\mathsf{ID} = (\mathsf{ID}_1, \cdots, \mathsf{ID}_l)$ at level l can use its own secret key d_{ID} to generate a secret key for any of its children $\mathsf{ID}' = (\mathsf{ID}_1, \cdots, \mathsf{ID}_l, \mathsf{ID}_{l+1})$ via $d_{\mathsf{ID}'} \leftarrow \mathsf{KeyDer}(d_{\mathsf{ID}}, \mathsf{ID}_{l+1})$. We refer to [4, 5, 7] for the security definitions for HIBE schemes.

2.3 Wildcarded Identity-based Encryption

We follow [9, 11] to define wildcarded identity-based encryption (WIBE) schemes with their security definitions. A pattern in our scheme is described by a vector $P = (P_1, \cdots, P_\ell) \in (\{0,1\}^* \cup \{*\})^\ell$, where $*$ is a special wildcard symbol. We say that a pattern $P' = (P'_1, \cdots, P'_{\ell'})$ belongs to P, denoted by $P' \in_* P$, if and only if in case $\ell' > \ell$ for all $i = 1, \cdots, \ell$, either $P'_i = P_i$ or $P_i = *$; in case $\ell' \leq \ell$ for all $i = 1, \cdots, \ell'$, either $P'_i = P_i$ or $P_i = *$ and $P_{\ell'+1} = \cdots = P_\ell = *$. We say P' matches P, denoted by $P' \approx P$, if and only if in case $\ell' \leq \ell$ for all $i = 1, \cdots, \ell'$, either $P'_i = P_i$ or $P_i = *$ or $P'_i = *$; in case $\ell' > \ell$ for all $i = 1, \cdots, \ell$, either $P'_i = P_i$ or $P_i = *$ or $P'_i = *$ and $P'_{\ell+1} = \cdots = P'_{\ell'} = *$.

If $P = (P_1, \cdots, P_l)$ is a pattern, then we define $W(P)$ to be the set of wildcard positions in P, i.e., $W(P) = \{1 \leq i \leq L : P_i = *\}$, and $\overline{W}(P)$ to be its complement. Hence $W(P) \cap \overline{W}(P) = \emptyset$ and $W(P) \cup \overline{W}(P) = \{1, \cdots, L\}$.

A WIBE scheme is a tuple consisting of five algorithms providing the following functionality.

$\mathsf{Setup}(1^\lambda, L, n)$ On input the security parameter λ, the maximum number of levels L and the maximum number of identities at each level n, it outputs the public parameters param and the master secret key msk.

$\mathsf{Extract}(\mathsf{param}, P, \mathsf{msk})$ On input the public parameters, a pattern P and the master secret key msk, it outputs the secret key sk_P for P.

$\mathsf{KeyDer}(\mathsf{param}, P', sk_P)$ On input the public parameters param, a pattern $P' = (P'_1, \cdots, P'_{l+1})$ and the secret key sk_P of a pattern P in which $P' \setminus \{P'_{l+1}\} \in_* P$, it generates the secret key $sk_{P'}$ for P'.

$\mathsf{Encrypt}(\mathsf{param}, P, \mathscr{M})$ On input the public parameters param, a pattern P and a message $\mathscr{M}$, it outputs a ciphertext C.

$\mathsf{Decrypt}(sk_{P'}, C, \mathsf{param})$ On input the secret key $sk_{P'}$ of a pattern P', a ciphertext C for pattern P, it outputs a message $\mathscr{M}$ if and only if $P' \approx P$.

For correctness, it requires that for all public parameter and master secret key pair $(\mathsf{param}, \mathsf{msk})$ output by Setup, all messages $\mathscr{M}$, and all patterns $P \in (\{0,1\}^* \cup \{*\})^\ell, P' \in (\{0,1\}^* \cup \{*\})^{\ell'}$ such that $P \approx P'$, we have

$$\mathsf{Decrypt}(\mathsf{KeyDer}(\mathsf{msk}, P), \mathsf{Encrypt}(\mathsf{param}, P', \mathscr{M})) = \mathscr{M}$$

with probability 1.

The security model of a WIBE scheme is similar to that of a HIBE scheme. Formally, the IND-WID-CPA security model is defined through the following game, played between an adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ and a challenger:

Setup The challenger generates $(\mathsf{param}, \mathsf{msk}) \leftarrow \mathsf{Setup}$ and sends param to $\mathscr{A}$.

Phase 1 The adversary runs $\mathscr{A}_1$ on param. The adversary is given access to a key derivation oracle that, on input of pattern $P \in (\mathbb{Z}_p^* \cup \{*\})^\ell$, returns the secret key $sk_P \leftarrow \mathsf{Extract}(\mathsf{param}, P, \mathsf{msk})$ corresponding to that identity.

Challenge The adversary outputs two equal-length messages $(\mathscr{M}_0, \mathscr{M}_1)$ and a challenge pattern P^* such that $P^* \not\approx P$ for all queried P, along with some state information *state*. The challanger chooses a bit $b \leftarrow \{0,1\}$, computes the ciphertext $C^* \leftarrow \mathsf{Encrypt}(\mathsf{param}, P^*, \mathscr{M}_b)$ and sends C^* to the adversary.

Phase 2 The adversary runs $\mathscr{A}_2$ on the input C^* and the state information *state*. The adversary is given access to a key derivation oracle as before, but not on any identity that matches P^*.

Guess The adversary outputs its guess $b' \in \{0,1\}$ for b.

The adversary wins the game if $b = b'$, and its advantage is defined as

$$\left| \Pr[\mathscr{A} \text{ wins}] - \frac{1}{2} \right|.$$

Definition 1. *A (t, q_K, ε)-adversary against the* IND-WID-CPA *security of the* WIBE *scheme is an algorithm that runs in time at most t, makes at most q_K key derivation oracle queries, and has advantage at least ε in the* IND-WID-CPA *game described above.*

We also define a weaker selective-identity (sWID) security notion, in which the adversary commits to the challenge pattern at the beginning of the game, before the public parameters are made available.

2.4 Computational Complexity Assumptions

Security of our system is based on the complexity assumption called the Modified-q-BDHE assumption [12], which is a simple modification of the well-known BDHE assumption [13].

Definition 2. ***Modified-q-*BDHE *problem:*** *Suppose that $(p, \mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T, e)$ is a bilinear group system, choose $y, \gamma, k, q \xleftarrow{\$} \mathbb{Z}_p$, a generator $g \in \mathbb{G}$. Given*

$$\vec{Y} = \left(g, g^y, \ldots, g^{y^q}, g^{y^{q+2}}, \ldots, g^{y^{2q}}, g^{y\gamma}, g^{y^2\gamma}, \ldots, g^{y^q\gamma}, g^{y^{q+2}\gamma}, \ldots, g^{y^{2q}\gamma}, g^k, g^{k(y\gamma+y)} \right)$$

it is difficult to distinguish between $T = e(g,\tilde{g})^{y^{q+1}k} \in \mathbb{G}_T$ or $T = R \xleftarrow{\$} \mathbb{G}_T$.

Let $\mathscr{A}$ be an adversary that solves the Modified-q-BDHE problem above, denote ε the advantage of $\mathscr{A}$:

$$\left| \Pr\left[\mathscr{A}(\vec{Y}, T = e(g,\tilde{g})^{y^{q+1}k}) = 0 \right] - \Pr\left[\mathscr{A}(\vec{Y}, T = R) = 0 \right] \right| \geq \varepsilon$$

Definition 3. *If there doesn't exist any polytime adversary who has a non-negligible advantage in solving the Modified-q-*BDHE *problem, then we say that the Modified-q-*BDHE *assumption holds.*

It is easy to see that to distinguish between $T = e(g,g)^{y^{q+1}k} \in \mathbb{G}_T$ or $T = R \xleftarrow{\$} \mathbb{G}_T$, one needs to have one of the values $g^{y^{q+1}}$ or $g^{y^{q+1}\gamma}$, however we do not have these elements in $\vec{Y}$ and there is also no way from $\vec{Y}$ to derive one of these elements.

3 Our proposed WIBE scheme

3.1 Construction

Our WIBE scheme is described as follows.

Setup$(1^\lambda, L, n)$: this algorithm takes as input the security parameter λ, the maximum number of levels L and the maximum number of identities in each level n. Let $\mathscr{ID} = \{ID_{i,j}\}_{\substack{i=1,\ldots,L \\ j=1,\ldots,n}}$ be the identity universe, and $\{ID_{i,0}\}_{i=1,\ldots,L}$ be a set of dummy identities. Let $\mathscr{H} : \{0,1\}^* \to \mathbb{G}$ be a target collision hash function. Assume that $D = (p, \mathbb{G}, \mathbb{G}_T, e)$ is a bilinear map group system, the algorithm randomly picks a generator $g \in \mathbb{G}$ and two scalars $x, y \in \mathbb{Z}_p^*$. The public parameters param and the master secret key msk are set:

$$\mathsf{param} = \left(D, g, g^y, , e(g,g)^x, \{\mathscr{H}(ID_{i,j})\}_{\substack{i=1,\ldots,L \\ j=0,\ldots,n}}\right); \quad \mathsf{msk} = g^x$$

In our scheme we require that all identities $\{ID_{i,j}\}_{\substack{i=1,\ldots,L \\ j=0,\ldots,n}}$ are different, and we consider msk as Root.

Extract(param, P, msk): to generate the secret key sk_P for a pattern $P = (P_1, \ldots, P_\ell), \ell \le L$, from the master secret key msk, the algorithm picks $t \xleftarrow{\$} \mathbb{Z}_p^*$ and computes $sk_P = (a,b,c,d)$ where

$$(a,b,c,d) = \left(g^x \cdot g^{yt}, g^t, \{\mathscr{H}(P_i)^t\}_{i \in \overline{W}(P)}, \{\mathscr{H}(\mathsf{ID}_{i,j})^t\}_{\substack{i \in W(P) \cup i = \ell+1,\ldots,L \\ j=0,\ldots,n}}\right)$$

Note that if P_i is not a wildcard then $P_i = \mathsf{ID}_{i,j}$ for some $j \in \{1,\ldots,n\}$. In our scheme, user only needs to keep a secret and can store (b,c,d) in the public domain, that mean the user's secret key in our scheme is of constant size.

KeyDer(param, P', sk_P): to generate the secret key $sk_{P'} = (a',b',c',d')$ for a pattern $P' = (P'_1, \ldots, P'_{\ell+1}), \ell + 1 \le L$ (note that $P'_{\ell+1}$ is not a dummy identity) and $P' \setminus \{P'_{\ell+1}\} \in_* P$, from the secret key $sk_P = (a,b,c,d)$, the algorithm picks $t'' \xleftarrow{\$} \mathbb{Z}_p^*$ and computes

$$(a',b',c',d') = \left(g^x \cdot g^{y(t+t'')}, g^{t+t''}, \{\mathscr{H}(P'_i)^{t+t''}\}_{i \in \overline{W}(P')}, \{\mathscr{H}(\mathsf{ID}_{i,j})^{t+t''}\}_{\substack{i \in W(P') \cup i = \ell+2,\ldots,L \\ j=0,\ldots,n}}\right)$$

Note that $a' = g^x \cdot g^{y(t+t'')} = a \cdot g^{yt''}$ and the algorithm has g^y from the param. In addition, since $P' \setminus \{P'_{\ell+1}\} \in_* P$, we have $W(P') \subseteq W(P) \cup \ell + 1$, so from d and t'' the algorithm is able to generate d'. Similarly, from c and t'' the algorithm is able to generate $\{\mathscr{H}(P'_i)^{t+t''}\}_{i \in \overline{W}(P)}$, and from d and t'' the algorithm is able to generate $\{\mathscr{H}(P'_i)^{t+t''}\}_{i \in \overline{W}(P') \setminus \overline{W}(P)}$, that means the algorithm is able to generate c'.

It is easy to see that $sk_{P'}$ is in the right form with the new randomness $t' = t + t''$, and user also only needs to keep a' secret.

$\mathsf{Encrypt}(\mathsf{param}, P, \mathscr{M})$: to encrypt a message $\mathscr{M} \in \mathbb{G}_T$ to a pattern $P = (P_1, \ldots, P_\ell)$, the algorithm first randomly picks $k \in \mathbb{Z}_p^*$, computes the ciphertext $C = (C_1, C_2, C_3)$, where:

$$C_1 = g^k, C_2 = \left(g^y \cdot \prod_{i \in \overline{W}(P)} \mathscr{H}(P_i) \prod_{i=\ell+1}^{L} \mathscr{H}(\mathsf{ID}_{i,0}) \right)^k, C_3 = \mathscr{M} \cdot e(g,g)^{kx}$$

Finally, it outputs $C = (C_1, C_2, C_3)$ which also includes the description of the pattern $P = (P_1, \ldots, P_\ell)$.

$\mathsf{Decrypt}(sk_{P'}, C, \mathsf{param})$: assume that $P = (P_1, \ldots, P_\ell)$ is the ciphertext pattern and $P' = (P'_1, \ldots, P'_\ell)$ is the key pattern. The algorithm first checks that whether or not $P' \approx P$, if not it returns $\perp$. Otherwise, since $P' \approx P$ that means $\forall i, P'_i = P_i$ or $P'_i = *$ or $P_i = *$, it is easy to see that c' and d' will contain $\{\mathscr{H}(P_i)^{t'}\}_{i \in \overline{W}(P)}$ and $\{\mathscr{H}(\mathsf{ID}_{i,0})^{t'})\}_{i=\ell+1}^{L}$, the algorithm hence first computes:

$$K = \frac{e(C_1, a' \cdot \prod_{i \in \overline{W}(P)} \mathscr{H}(P_i)^{t'} \prod_{i=\ell+1}^{L} \mathscr{H}(\mathsf{ID}_{i,0})^{t'})}{e(b', C_2)}$$

$$= \frac{e(g^k, g^x \cdot g^{yt'} \cdot \prod_{i \in \overline{W}(P)} \mathscr{H}(P_i)^{t'} \prod_{i=\ell+1}^{L} \mathscr{H}(\mathsf{ID}_{i,0})^{t'})}{e(g^{t'}, (g^y \cdot \prod_{i \in \overline{W}(P)} \mathscr{H}(P_i) \prod_{i=\ell+1}^{L} \mathscr{H}(\mathsf{ID}_{i,0}))^k)} = e(g,g)^{kx}$$

then recovers the message

$$\mathscr{M} = C_3 / K$$

3.2 Security

Theorem 4. *Assume that $q \geq (n+1)L$, our* WIBE *scheme is selectively secure under the Modified-q-*BDHE *assumption.*

Proof. Let $\mathscr{S}$ and $\mathscr{A}$ be adversaries who attack Modified-q-BDHE assumption and our WIBE scheme, respectively. We will prove that $\mathscr{S}$ can play the role of a simulator to simulate $\mathscr{A}$ and then rely on $\mathscr{A}$'s output to break the security of Modified-q-BDHE assumption.

First, $\mathscr{A}$ sends the target pattern $P^* = (P_1^*, \ldots, P_{\ell^*}^*)$ to $\mathscr{S}$. Note that $\mathscr{S}$ also has an instance of Modified-q-BDHE assumption, as well as the maximum number of levels L and the maximum number of identities in each level n.

Setup: $\mathscr{S}$ first considers a boolean formula with AND-gates:

$$\mathscr{P} = (\wedge P_i^*)_{i \in \overline{W}(P^*)} \wedge (\wedge \mathscr{H}(\mathsf{ID}_{i,0}))_{i=\ell^*+1,\ldots,L}$$

then uses the algorithm in Appendix G in [14] to build a LSSS matrix $(M_{\ell \times \ell'}, \rho)$ from $\mathscr{P}$, where $\ell = |\overline{W}(P^*)| + L - \ell^*$ ($|\overline{W}(P^*)|$ is the cardinality of set $\overline{W}(P^*)$) and $\ell, \ell' \leq q$.

Note that $\{\overline{W}(P^*)\} \cup \{\ell^*+1, \ldots, L\} = \{\rho(i)\}_{i=1,\ldots,\ell}$, ρ is an injective function.

$\mathscr{S}$ next chooses $x' \xleftarrow{\$} \mathbb{Z}_p^*$ and implicitly sets $x = x' + y^{q+1}$, generates $e(g,g)^x = e(g,g)^{x'} \cdot e(g^y, g^{y^q})$.

Denote $h_1 = \mathscr{H}(ID_{1,1}), h_2 = \mathscr{H}(ID_{1,2}), \ldots, h_{n+1} = \mathscr{H}(ID_{2,1}), \ldots, h_N = \mathscr{H}(ID_{L,n}), N = nL$. Denote $h_{N+1} = \mathscr{H}(ID_{1,0}), \ldots, h_{N+L} = \mathscr{H}(ID_{L,0})$

To generate $\{h_i\}_{i=1,\ldots,N+L}$, $\mathscr{S}$ implicitly sets

$$\overrightarrow{\tau} = (\gamma, \gamma y, \gamma y^2, \ldots, \gamma y^{\ell'-1})^\perp \in \mathbb{Z}_p^{\ell'}$$

Let $\overrightarrow{\lambda} = M \cdot \overrightarrow{\tau}$ be the vector shares, for all $j = 1, \ldots, \ell$ we have

$$\lambda_j = \sum_{i \in [\ell']} M_{j,i} \gamma y^{i-1}$$

Although $\mathscr{S}$ cannot compute $\overrightarrow{\lambda}$, $\mathscr{S}$ is able to find constants $\{\omega_i\}_{1 \leq i \leq \ell}$ satisfying:

$$\sum_{i=1,\ldots,\ell} \omega_i \cdot \lambda_i = \gamma$$

Note that, based on the property of LSSS matrix, $\mathscr{S}$ is able to find $\{\omega_i\}_{1 \leq i \leq \ell}$ satisfying (M_i is row i of matrix M)

$$\sum_{i=1,\ldots,\ell} \omega_i \cdot M_i = (1, 0, \ldots, 0)$$

or

$$\sum_{i=1,\ldots,\ell} \omega_i \cdot M_{i,1} \gamma = \gamma$$

It is then easy to deduce that

$$\sum_{i=1,\ldots,\ell} \omega_i \cdot \lambda_i = \gamma$$

On the other hand, $\mathscr{S}$ has $g^{\gamma y^i}, i = 1, \ldots, \ell'$ from the assumption (note that $q \geq N = nL \geq \ell'$) and matrix M, $\mathscr{S}$ thus can generate $\{h_i\}_{i=1,\ldots,N+L}$ by: first, for each h_j where there exists $i \in [\ell]$ satisfying $h_j = \mathscr{H}(P^*_{\rho(i)})$ or $h_j = \mathscr{H}(\mathsf{ID}_{\rho(i),0})$ (ρ is an injective function), $\mathscr{S}$ picks randomly scalar $z_j \xleftarrow{\$} \mathbb{Z}_p$ then computes

$$h_j = g^{z_j} \cdot g^{\omega_i \sum_{a \in [\ell']} M_{i,a} \gamma y^a} = g^{z_j} \cdot g^{y \omega_i \lambda_i}$$

We emphasize here the relation between indices i and j is that $h_j = \mathscr{H}(P^*_{\rho(i)})$ or $h_j = \mathscr{H}(\mathsf{ID}_{\rho(i),0})$.

Second, for each h_j, where there doesn't exists $i \in [\ell]$ satisfying $h_j = \mathscr{H}(P^*_{\rho(i)})$ or $h_j = \mathscr{H}(\mathsf{ID}_{\rho(i),0})$, $\mathscr{S}$ picks randomly scalar $z_j \xleftarrow{\$} \mathbb{Z}_p$ then computes $h_j = g^{z_j}$.

Since all z_j are randomly chosen, $h_1, \ldots, h_{N+L}$, are in the right form.

$\mathscr{S}$ eventually sets param as

$$\mathsf{param} = (g, g^y, e(g,g)^x, h_1, \ldots, h_{N+L})$$

and sends it to $\mathscr{A}$, note that $h_1 = \mathscr{H}(ID_{1,1}), h_2 = \mathscr{H}(ID_{1,2}), \ldots, h_{n+1} = \mathscr{H}(ID_{2,1}), \ldots, h_N = \mathscr{H}(ID_{L,n}), N = nL$ and $h_{N+1} = \mathscr{H}(ID_{1,0}), \ldots, h_{N+L} = \mathscr{H}(ID_{L,0})$.

Query phase 1: $\mathscr{A}$ issues $q_1, \ldots, q_{K_1}$ queries as follows:

1. $\mathscr{A}$ requests to know the secret key associated with the pattern $P = (P_1, \ldots, P_v)$ where $P \not\approx P^*$;
2. $\mathscr{A}$ requests to know a part of the secret key associated with the pattern $P = (P_1, \ldots, P_v)$ where $P \approx P^*$. The reason why $\mathscr{A}$ is able to make this query is that this part of the secret key is stored in the public domain.

To answer the first one, $\mathscr{S}$ first finds the set of indices R' such that $\forall i \in R, (P_i \neq P^*_i) \wedge (P_i \neq *) \wedge (P^*_i \neq *)$, and sets $R = R' \cup \{\ell^* + 1, \ldots, v\}$ (note that in case $v \leq \ell^*$ the set $R = R'$). Since $P \not\approx P^*$, the set $R \neq \emptyset$. Let $S = (\overline{W}(P^*) \cup \{\ell^* + 1, \ldots, L\}) \setminus R$.

To be more clear, we give two examples corresponding to the cases $v \leq \ell^*$ and $v > \ell^*$. Assume $P = (\mathsf{ID}_{1,3}, \mathsf{ID}_{2,5}, *, \mathsf{ID}_{4,2}, \mathsf{ID}_{5,8})$ and $P^* = (\mathsf{ID}_{1,2}, \mathsf{ID}_{2,5}, \mathsf{ID}_{3,4}, *, \mathsf{ID}_{5,8}, \mathsf{ID}_{6,1})$. In this case, $R = 1$ and $S = 2, 3, 5, 6, 7, \ldots, L$.

For the second example, assume $P = (\mathsf{ID}_{1,3}, \mathsf{ID}_{2,5}, *, \mathsf{ID}_{4,2}, \mathsf{ID}_{5,8}, \mathsf{ID}_{6,1})$ and $P^* = (\mathsf{ID}_{1,3}, \mathsf{ID}_{2,5}, \mathsf{ID}_{3,4}, *, \mathsf{ID}_{5,8})$. In this case, $R = 6$ and $S = 1,2,3,5,7,8,\ldots,L$.

According to the well-known partition technique proof, we now need to program in such a way that $\mathscr{S}$ can compute the secret key for P but cannot compute the secret key for P^* (to avoid the trivial attack).

To this aim, $\mathscr{S}$ constructs a vector $\overrightarrow{u} = (u_1, \ldots, u_{\ell'}) \in \mathbb{Z}_p^{*\ell'}$ where $u_1 = -1$ and $\forall i = 1, \ldots, \ell$ where $\rho(i) \in S$, the product $\langle \overrightarrow{u} \cdot M_i \rangle = 0$. We notice here that such vector exists due to the property of LSSS matrix, . To continue, $\mathscr{S}$ picks $r \xleftarrow{\$} \mathbb{Z}_p$ then implicitly set:

$$t = r + u_1 y^q + u_2 y^{q-1} + \cdots + u_{\ell'} y^{q-\ell'+1}$$

Next, generates the secret key $sk_P = (a, b, c, d)$:

$$a = g^{x'} g^{yr} \prod_{i=2,\ldots,\ell'} (g^{y^{q+1-i}})^{u_i} = g^x \cdot g^{y \cdot t}$$

Note that $u_1 = -1$, it is easy to see that the unknown term $g^{y^{q+1}}$ in g^x is canceled out, since $g^{y \cdot t}$ contains $g^{-y^{q+1}}$. On the other hand, $\mathscr{S}$ knows vector $\overrightarrow{u}$, so he/she can generate:

$$b = g^t = g^r \prod_{i=1,\ldots,\ell'} (g^{y^{q+1-i}})^{u_i}$$

Next, $\mathscr{S}$ needs to compute $c = \{\mathscr{H}(P_i)^t\}_{i \in \overline{W}(P)}$. To this aim, for the set $\{P_i \neq P_i^*\}_{i \in \overline{W}(P)}$ (that means there doesn't exists $i' \in [\ell]$ satisfying $\mathscr{H}(P_i) = \mathscr{H}(P^*_{\rho(i')})$, note that identities in all level are different), $\mathscr{S}$ knows z_j such that $h_j = g^{z_j} = \mathscr{H}(P_i)$, therefore $\mathscr{S}$ simply computes:

$$\mathscr{H}(P_i)^t = (g^t)^{z_j}$$

For the set $\{P_i = P_i^*\}_{i \in \overline{W}(P)}$ (that means there exists $i' \in [\ell]$ satisfying $\mathscr{H}(P_i) = \mathscr{H}(P^*_{\rho(i')})$), we have from the setup phase that

$$\mathscr{H}(P_i) = h_j = g^{z_j} \cdot g^{\omega_{i'} \sum_{a \in [\ell']} M_{i',a} \gamma y^a}$$

$\mathscr{S}$ knows z_j and computes

$$\mathscr{H}(P_i)^t = (g^t)^{z_j} \cdot g^{(r + u_1 y^q + u_2 y^{q-1} + \cdots + u_{\ell'} y^{q-\ell'+1}) \omega_{i'} \sum_{a \in [\ell']} M_{i',a} \gamma y^a}$$

The key point for $\mathscr{S}$ to compute $\mathscr{H}(P_i)^t$ is that the product $\langle \overrightarrow{u} \cdot M_{i'} \rangle = 0$, that means $\mathscr{S}$ doesn't need to know the unknown term $\tilde{g}^{y^{q+1}\gamma}$. For other terms $\mathscr{S}$ has already known from the assumption.

Note that, in case $v \leq \ell^*$, $\mathscr{S}$ cannot compute the secret key for P^* since he/she cannot compute the values

$$\{\mathscr{H}(P_i^*)^t\}_{i \in R}$$

since $\langle \overrightarrow{u} \cdot M_{i'} \rangle \neq 0, \rho(i') = i$. For the above first example, $\mathscr{S}$ cannot compute the value $\mathscr{H}(\mathsf{ID}_{1,2})^t$. That is exactly the well-known partition technique proof.

Finally, $\mathscr{S}$ needs to compute

$$d = \{\mathscr{H}(\mathsf{ID}_{i,j})^t\}_{\substack{i \in W(P) \cup i = v+1,\ldots,L \\ j = 0,\ldots,n}}$$

it is similar to the case of computing c.

Note that in case $v > \ell^*$, $\mathscr{S}$ cannot compute the secret key for P^* since he/she cannot compute the values

$$\{\mathscr{H}(\mathsf{ID}_{i,0})^t\}_{i \in \{\ell^*+1,\ldots,v\}}$$

since $\langle \overrightarrow{u} \cdot M_{i'} \rangle \neq 0, \rho(i') = i$. For the above second example, $\mathscr{S}$ cannot compute the value $\mathscr{H}(\mathsf{ID}_{6,0})^t$. That is exactly the well-known partition technique proof.

To answer the second type of query, note that the unknown element g^x only appear in a. However, fortunately $\mathscr{S}$ just needs to provide to $\mathscr{A}$ the values (b,c,d), this leads to the fact that $\mathscr{S}$ can simply picks $t \xleftarrow{\$} \mathbb{Z}_p^*$ then generates (b,c,d) and returns to $\mathscr{A}$.

Challenge: The simulator $\mathscr{S}$ computes the challenge ciphertext $C^* = (C_1^*, C_2^*, C_3^*)$ as follows:

First, finds all the known values z_j such that

$$h_j = \mathscr{H}(P^*_{\rho(i)}) = g^{z_j} \cdot g^{\omega_i \sum_{a \in [\ell']} M_{i,a} \gamma y^a} = g^{z_j} \cdot g^{y \omega_i \lambda_i}$$

or

$$h_j = \mathscr{H}(\mathsf{ID}_{\rho(i),0}) = g^{z_j} \cdot g^{\omega_i \sum_{a \in [\ell']} M_{i,a} \gamma y^a} = g^{z_j} \cdot g^{y \omega_i \lambda_i}$$

Let S' be the set of all such above indices j. Note that $S' = \{\overline{W}(P^*)\} \cup \{\ell^* + 1, \ldots, L\}, |S'| = \ell$, and the relation between indices i and j is that $h_j = \mathscr{H}(P^*_{\rho(i)})$, or $h_j = \mathscr{H}(\mathsf{ID}_{\rho(i),0}), i \in [\ell]$, and we have $\sum_{i=1,\ldots,\ell} \omega_i \cdot \lambda_i = \gamma$ from setup phase.

$\mathscr{S}$ computes (note that he/she knows $g^k, g^{k(y+y\gamma)}$ from the assumption)

$$\begin{aligned} C_1^* = g^k, C_2^* &= \left(g^{k(y+y\gamma)} g^{\sum_{j \in S'} k z_j}\right) = \left(g^y \cdot \prod_{j \in S'} g^{z_j} \cdot \prod_{i \in [\ell]} g^{y \omega_i \lambda_i}\right)^k = \left(g^y \prod_{j \in S'} h_j\right)^k \\ &= \left(g^y \prod_{i \in \overline{W}(P^*)} \mathscr{H}(P_i^*) \prod_{i=\ell^*+1}^{L} \mathscr{H}(\mathsf{ID}_{i,0})\right)^k \end{aligned}$$

To compute C_3^*, $\mathscr{S}$ first picks randomly a bit $b \in \{0,1\}$ and computes

$$C_3^* = \mathscr{M}_b \cdot T \cdot e(g^k, g)^{x'}$$

It is easy to see that in case $T = e(g,g)^{y^{q+1}k}$ then it is easy to see that C_3^* is in valid form. Otherwise, C_3^* is a random element in $\mathbb{G}_T$.

Query phase 2: $\mathscr{A}$ issues $q_{K_1+1}, \ldots, q_K$ queries similarly to Phase 1

Guess: Eventually, $\mathscr{A}$ returns his/her bit guess b' for b. $\mathscr{S}$ checks that if $b' = b$, he/she outputs 0 to guess that $T = e(g,g)^{y^{q+1}k}$. Otherwise, $\mathscr{S}$ outputs 1, which means that T is a random element in $\mathbb{G}_T$.

When $T = e(g,g)^{y^{q+1}k}$ the simulator $\mathscr{S}$ gives a perfect simulation so we have that

$$\Pr\left[\mathscr{S}(\vec{Y}, T = e(g,g)^{y^{q+1}k}) = 0\right] = \frac{1}{2} + \varepsilon$$

When T is a random group element the message $\mathscr{M}_b^*$ is completely hidden from the adversary and we have

$$Pr[\mathscr{S}(\vec{Y}, T = R) = 0] = \frac{1}{2}$$

Therefore, the simulator can play the Modified-q-BDHE game with non-negligible advantage (equal to ε) or she can break the security of Modified-q-BDHE assumption, which concludes our proof. □

4 Performance Analysis

We give in the Table 1 the comparison table among our scheme and the scheme at ESORICS'18 as well as some other relevant WIBE schemes proposed in the literature.

From this table, we see that only our scheme and the recent scheme at ESORICS'18 [11] support wildcard at KeyDer algorithm (generating secret key and delegating secret key). Regarding the storage, user in our scheme only needs to store one element (about 170 bits) while user in other existing schemes needs to store at least L elements (about $L \cdot 170$ bits), where L is the maximum number of levels. In addition, ciphertext size in our scheme is also shorter than ciphertext size in ESORICS'18 scheme (less than one element). Regarding the computation complexity, only in our scheme both encryption time and decryption time are independent to L (we omit the multiplication operation since it is very fast when comparing to exponential operation and pairing operation). More precisely, at encryption step we need 3 exponential operations and at decryption step we need two paring operations. Overall, our scheme is very suitable for lightweight device-based applications such as IoTs applications.

Two shortcomings of our scheme are the large public storage and the maximum number of identities needs to be fixed at the setup. However, the former may not be big problems since the public parameters may not need to be stored permanently on the client, and can be accessed on demand from a non-weak server with large computational resources. Moreover, keys are typically much more smaller than the data and/or its ciphertext in real-life scenarios.

Table 1: Comparison of WIBE schemes. Here L is the maximum number of levels, $|sk|$ and $|C|$ are the secret key size and ciphertext size respectively, n is the length of an identity string, e is the time of exponential operation, p is the time of pairing. In the scheme column for the WIBE schemes in [9], we include also the corresponding HIBE schemes. We note that multiplication operation is very fast when we compare with the exponential operation and pairing operation, we thus omit it in the comparison table.

Scheme	Wildcard use	|sk|	|C|	Decrypt	Encrypt
[9]+[6]	Enc	L+1	2L+2	(L+1)p	(L+3)e+p
[9]+[7]	Enc	L+2	L+3	Le+2p	(L+3)e+p
[9]+[8]	Enc	L+1	(n+1)L+2	(L+1)p	(L+3)e+p
[11]	Enc+KeyDer	2L+3	4	Le+3p	(L+3)e+p
Ours	Enc+KeyDer	**1**	**3**	**2p**	**3e**

5 Conclusion

In this paper, we propose a novel technique to design an efficient wildcarded Identity-based Encryption scheme, which improves the current most efficient one by Kim et al. [11] at ESORICS 2018. Our scheme has constant size ciphertext (consisting of 3 group elements) and secret key (consisting of only 1 group element), and fast decryption (computing two Parings). Our scheme can be transformed to obtain CCA security using the transformations in [10] which utilized the CHK transformation [15] and a transform of Dent [16]. Two drawback of our scheme are that it requires larger public storage than existing schemes and is not scalable. We will leave as a future work the problem on how to resolve such issues.

Despite that there have been several construction of HIBE [17, 18] and ABE [19, 20] from lattices, a construction for lattice-based WIBE has not been yet available. We will leave this also as a future work.

Acknowledgments

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.301.

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of the Advances in Cryptology (CRYPTO'84), Santa Barbara, California, USA*, ser. Lecture Notes in Computer Science, vol. 196. Springer, Berlin, Heidelberg, August 1984, pp. 47–53.

[2] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Proc. of the 21st Annual International Cryptology Conference (CRYPTO'01), Santa Barbara, California, USA*, ser. Lecture Notes in Computer Science, vol. 2139. Springer, Berlin, Heidelberg, August 2001, pp. 213–229.

[3] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, August 2003.

[4] C. Gentry and A. Silverberg, "Hierarchical id-based cryptography," in *Proc. of the 8th International Conference on the Theory and Application of Cryptology and Information Security(ASIACRYPT'02), Queenstown, New Zealand*, ser. Lecture Notes in Computer Science, vol. 2501. Springer, Berlin, Heidelberg, December 2002, pp. 548–566.

[5] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02), Amsterdam, The Netherlands*, ser. Lecture Notes in Computer Science, vol. 2332. Springer, Berlin, Heidelberg, April-May 2002, pp. 466–481.

[6] D. Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles," in *Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04), Interlaken, Switzerland*, ser. Lecture Notes in Computer Science, vol. 3027. Springer, Berlin, Heidelberg, May 2004, pp. 223–238.

[7] D. Boneh, X. Boyen, and E. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'05), Aarhus, Denmark*, ser. Lecture Notes in Computer Science, vol. 3494. Springer, Berlin, Heidelberg, May 2005, pp. 440–456.

[8] B. Waters, "Efficient identity-based encryption without random oracles," in *Proc. of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'05), Aarhus, Denmark*, ser. Lecture Notes in Computer Science, vol. 3494. Springer, Berlin, Heidelberg, May 2005, pp. 114–127.

[9] M. Abdalla, D. Catalano, A. W. Dent, J. Malone-Lee, G. Neven, and N. P. Smart, "Identity-based encryption gone wild," in *Proc. of the 33rd International Colloquium on Automata, Languages, and Programming (ICALP'06), Venice, Italy*, ser. Lecture Notes in Computer Science, vol. 4052. Springer, Berlin, Heidelberg, July 2006, pp. 300–311.

[10] M. Abdalla, J. Birkett, D. Catalano, A. W. Dent, J. Malone-Lee, G. Neven, J. C. N. Schuldt, and N. P. Smart, "Wildcarded identity-based encryption," *Journal of CRYPTO*, vol. 24, no. 1, pp. 42–82, February 2011.

[11] J. Kim, S. Lee, J. Lee, and H. Oh, "Scalable wildcarded identity-based encryption," in *Proc. of the 23rd European Symposium on Research in Computer Security (ESORICS'18), Barcelona, Spain*, ser. Lecture Notes in Computer Science, vol. 11099. Springer, Berlin, Heidelberg, September 2018, pp. 269–287.

[12] Q. M. Malluhi, A. Shikfa, and V. C. Trinh, "A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption," in *Proc. of ACM Asia Conference on Computer and Communications Security (AsiaCCS'17), Abu Dhabi, United Arab Emirates*, ser. ASIA CCS '17. Association for Computing Machinery, April 2017, pp. 230–240.

[13] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. of the 25th Annual International Cryptology Conference (CRYPTO'05), Santa Barbara,*

California, USA, ser. Lecture Notes in Computer Science, vol. 3621. Springer, Berlin, Heidelberg, Augusut 2005, pp. 258–275.

[14] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'11), Tallinn, Estonia*, ser. Lecture Notes in Computer Science, vol. 6632. Springer, Berlin, Heidelberg, May 2011, pp. 568–588.

[15] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04), Interlaken, Switzerland*, ser. Lecture Notes in Computer Science, vol. 3027. Springer, Berlin, Heidelberg, May 2004, pp. 207–222.

[16] A. W. Dent, "A designer's guide to kems," in *Proc. of the 9th International Conference on Cryptography and Coding (IMA'03), Cirencester, UK*, ser. Lecture Notes in Computer Science, vol. 2898. Springer, Berlin, Heidelberg, December 2003, pp. 133–151.

[17] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (h)ibe in the standard model," in *Proc. of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10), Monaco and Nice, France*, ser. Lecture Notes in Computer Science, vol. 6110. Springer, Berlin, Heidelberg, June 2010, pp. 553–572.

[18] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Proc. of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10), Monaco and Nice, France*, ser. Lecture Notes in Computer Science, vol. 6110. Springer, Berlin, Heidelberg, June 2010, pp. 523–552.

[19] X. Boyen, "Attribute-based functional encryption on lattices," in *Proc. of the 10th Theory of Cryptography Conference (TCC'13), Tokyo, Japan*, ser. Lecture Notes in Computer Science, vol. 7785. Springer, Berlin, Heidelberg, 2013, pp. 122–142.

[20] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Attribute-based encryption for circuits," *Journal of the ACM*, vol. 62, no. 6, pp. 45:1–45:33, December 2015.

Author Biography

Dung Hoang Duong is a lecturer in the School of Computing and Information Technology at University of Wollongong since 2018. He received a Ph.D. degree in mathematics from Leiden University, the Netherlands, in 2013. He was a postdoctoral fellow in the Faculty of Mathematics at Bielefeld University from 2013 to 2015. From 2015 to 2018, he was an assistant professor at the Insitute of Mathematics for Industry at Kyushu University, Japan. His research interests are post-quantum cryptography, especially lattice-based cryptography and multivariate public key cryptography.

Willy Susilo is a Senior Professor in the School of Computing and Information Technology, Faculty of Engineering and Information Sciences at the University of Wollongong (UOW), Australia. He is the director of Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, UOW and the Head of School of Computing and Information Technology at UOW (2015 - now). Prior to this role, he was awarded the prestigious Australian Research Council Future Fellowship in 2009. In 2016, he was awarded the "Researcher of the Year? at UOW, due to his research excellence and contributions. He is the Editor-in-Chief of the Elsevier's Computers Standards and Interface and the Information journal. He is currently an Associate Editor of IEEE Transactions on

Dependable and Secure Computing. He has also served as the program committee member of several international conferences.

Viet Cuong Trinh received the Ph.D. degrees in computer science from Paris 8 University, Paris, France, in 2013. Before joining Hong Duc University, Viet Nam, in 2016, he was a postdoctoral researcher with the Orange Labs, Caen, France and then KINDI research center, Qatar University, Doha, Qatar. He is currently an Assistant Professor with the Faculty of Information and Communication Technology, Hong Duc University, Viet Nam. His research is focused on cryptography, in particular on provable security for cryptographic schemes.