Cryptanalysis and Improvement on Wang *et al.*'s Attribute-Based Searchable Encryption Scheme

Yi-Fan Tseng, Chun-I Fan, Si-Jing Wu, Hsin-Nan Kuo, and Jheng-Jia Huang

***Abstract*—Searchable encryption is a powerful and useful primitive when users want to store their encrypted files on cloud storages. In this paper, we demonstrate security flaws of the searchable encryption scheme proposed by Wang *et al*. in 2017. Furthermore, we propose a solution to fix the flaws, and the improved scheme also largely reduces the length of the ciphertext such that it is independent of the number of the attributes.**

***Index Terms*—Attribute-based encryption, cryptanalysis, hidden policy, searchable encryption.**

I. Introduction

For the increasingly expanding implementation of cloud computing, more and more users upload their data to cloud servers. Since classified data are outsourced to cloud servers, data owners often concern about the privacy of such data. How to gain the confidentiality of cloud data while making the fullest use of secure cloud systems becomes an important issue in cloud computing. Typically, a data owner encrypts private data before outsourcing to cloud server. Thus, how to search such encrypted files in the cloud through keywords is another essential issue. A well-known solution to mitigate the aforementioned issue is to deploy searchable encryption (SE) [1] which enables cloud servers to search encrypted data without leaking any information either of the keyword or the plaintext data.

In 2017, Wang *et al*. [2] proposed an attribute-based encryption with keyword search (ABKS) [3]-[6] scheme based on ciphertext-policy attribute-based encryption (CP-ABE), which preserves the fine-grained access control inherited from the ABE system. However, we found that there are some problems in the scheme. The length of a ciphertext is not independent of the number of attributes. Moreover, neither the privacy of attributes nor keywords is preserved. In order to deal with these problems, we give the cryptanalysis and present an improvement on their scheme.

Manuscript received May 15, 2019; revised August 5, 2019.

Yi-Fan Tseng is with the Department of Computer Science, National Chengchi University, Taipei, Taiwan (e-mail: yftseng@cs.nccu.edu.tw).

Chun-I Fan is with the Telecom Technology Center, Department of Computer Science and Engineering, National Sun Yat-sen University, Information Security Research Center, National Sun Yat-sen University, and Intelligent Electronic Commerce Research Center, National Sun Yat-sen University, Kaohsiung, Taiwan (Corresponding author; e-mail: cifan@mail.cse.nsysu.edu.tw).

Si-Jing Wu and Hsin-Nan Kuo are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan (e-mail: jim5566556@gmail.com, bluedunk@gmail.com).

Jheng-Jia Huang is with the Telecom Technology Center, Kaohsiung, Taiwan (e-mail: d013040001@g-mail.nsysu.edu.tw).

II. Preliminaries

In this section, we briefly review some backgrounds essential to understand the proposed work.

A. Multilinear Maps

Let $\mathbb{G}_1, .., \mathbb{G}_n$ and $\mathbb{G}_\mathrm{T}$ be the prime ordered (say, p) cyclic groups. A cryptographic n-linear map [7] is defined as

$$e: \mathbb{G}_1 \times \cdots \times \mathbb{G}_n \to \mathbb{G}_T$$

for $n > 2$ that satisfies the following properties:

1. **Multilinearity:** The condition of
$$e\left(g_1^{x_1}, \dots, g_n^{x_n}\right) = e(g_1, \dots, g_n)^{\prod_{i=1}^{n} x_i}$$
always holds for $a_i \in Z_p^*$ and $g_i \in_R \mathbb{G}_i$, where $i \in [1, n]$.

2. **Non-degeneracy:** For a random generator $g_i \in \mathbb{G}_i$, the condition
$$e(g_1, \dots, g_n) = g_T$$

always holds where g_T is the generator of $\mathbb{G}_T$.

3. **Computability:** There should be an efficient algorithm to compute the map e.

An n-linear map e is called symmetric n-linear map if

$$\mathbb{G}_1 = \mathbb{G}_2 = \cdots = \mathbb{G}_n.$$

Therefore, a symmetric n-linear map can be defined as

$$e: \mathbb{G}_1 \times \cdots \times \mathbb{G}_1 \to \mathbb{G}_\mathrm{T}.$$

Besides, for an n-linear map, there is a set of bilinear maps

$$e_{i,j}: \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j}, \forall i, j > 0 \text{ and } i + j \leq n.$$

B. Access Structure

In the proposed scheme, we use a series of "AND gate" on multi-value attribute as the underline access structure. Let n be the total number of attributes. We consider

$$S = \{s_1, s_2, \cdots, s_n\}$$

as the universe attribute list, and for an attribute $s_i \in S$ we set

$$T_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,n_i}\}$$

where j is the number of the possible values for s_i. Suppose,

$$A = \{x_1, x_2, \cdots, x_n\}$$

is the attribute list of any user where $x_i \in T_i$, and

$$U = \{u_1, u_2, \cdots, u_n\}$$

is an access structure in the ciphertext where $u_i \in T_i$. We mention that the user's attribute list A satisfies the access

doi: 10.18178/ijmlc.2019.9.5.851

policy S if and only if $x_i = u_i$, $\forall\ i \in [1, \cdots, n]$.

C. Attribute-Based Encryption with Keyword Search

Attribute-based encryption with keyword search (ABKS) is an extended cryptographic primitive from attribute-based encryption. There have been lots of research on ABKS [8]-[10]. We review the definition for attribute-based encryption and authorized keyword search in the section.

Attribute-Based Encryption

For the sake of protecting encrypted information, most applications use complex access control mechanisms. Due to the fine-grained access control policy, there has been a great deal of interest in studying attribute-based encryption, and many related schemes have been proposed [11], [12]. In 2005, introduced by Sahai and Waters [13], attribute-based encryption (ABE), which is regarded as an extension of identity-based encryption (IBE), enables users to implement fine-grained access controls on the encrypted sensitive data. However, the scheme is lack of expressiveness. In order to make ABE more efficient and more flexible, Goyal *et al.* [14] in 2006 presented two different types of ABE schemes: key-policy ABE (KP-ABE) [15], [16] and ciphertext-policy ABE (CP-ABE) [17], [18]. In KP-ABE schemes, each user's private key is related to an access policy, and a ciphertext is associated with a set of attributes. A secret key can decrypt a ciphertext if and only if the attribute set associated with the ciphertext satisfies the access policy related to the user's private key. The situation in CP-ABE schemes is inversed.

To apply ABE schemes in terminal devices, lots of researches were proposed in literatures [19], [20]. In 2007, Cheung and Newport [21] implemented Boolean function, i.e., AND gate. in the standard model. However, the scheme cannot achieve the feature of hidden access policy. To be suitable for the data-outsourced cloud environment, Yu *et al.* [22] in 2010 adopted lazy re-encryption and proxy re-encryption. Several related ABE schemes can be found in references [23]-[25].

A CP-ABE scheme includes the following four algorithms:

- **Setup** $(1^{\lambda}) \rightarrow (PK, MK)$

The private key generator (PKG) takes a security parameter λ as an input. It outputs a public key PK and a master secret key MK.

- **KeyGen** $(PK, MK, U) \rightarrow SK_U$

On inputting the public key PK, the master secret key MK and the attribute set of user U, it outputs a user private key SK_U.

- **Encrypt** $(PK, M, \mathbb{S}) \rightarrow CT_{\mathbb{S}}$

It takes the public key PK, the message M and the access structure $\mathbb{S}$ as input, and outputs a ciphertext $CT_{\mathbb{S}}$.

- **Decrypt** $(CT_{\mathbb{S}}, SK_U) \rightarrow M$

The decryptor takes the ciphertext $CT_{\mathbb{S}}$ and the user private key SK_U as inputs, and returns a message M.

These algorithms must satisfy the correctness condition, i. e., for $SK_U \leftarrow (PK, MK, U)$ and $CT_{\mathbb{S}} \leftarrow$ Encrypt $(PK, M, \mathbb{S})$, one can decrypt the ciphertext as $M \leftarrow$Decrypt $(CT_{\mathbb{S}}, SK_U)$.

Authorized Keyword Search

To avoid leaking the information of keywords while tracking over the encrypted data, Boneh *et al.* [26] in 2004 proposed the concept of public key encryption with keyword search, but the scheme failed to achieve fine-grained access control on encrypted files. In 2014, Sun [27] and Zheng [28] independently presented ABKS schemes so as to resolve the problem. However, the size of the ciphertext is related to the number of attributes, so that the schemes have high computational costs. In 2015, Zheng *et al.* [29] proposed a certificateless keyword search scheme, but the scheme does not ensure the authority of search results.

In order to improve the computational time and above problems, Li *et al.* [30] in 2015 presented the authentication search scheme and made the application scene more flexible. Following Li *et al.*'s work, Lee *et al.* [31] in 2016 implemented hash table on searchable encryption. To protect the privacy information in the ciphertext, Li *et al.* [32] in 2017 presented a scheme which supports partially hidden access structures. More recent researches can be found in references [33]-[35].

III. Related Work

In this section we review some ABKS schemes [27], [38], [39] which we will compare our improved scheme with. Since we will demonstrate the comparison in secret key and ciphertext size, we show only **Setup**, **KeyGen**, **Encrypt, Decrypt** algorithms here. Readers are referred to [27], [38], [39] for further details. Besides, g is used to denote the generator of the source pairing $\mathbb{G}$, where $|\mathbb{G}| = p$ is a large prime.

A. Sun et al.'s Scheme [27]

Setup(1^{λ}): Taking the security parameter as input, the algorithm first chooses $y, t_1, t_2, \dots, t_n \in \mathbb{Z}_p$ as the master secret key MK, where n is the size of the attribute universe $\mathcal{U}$. The system parameter of the scheme is then generated as $PK = (g, Y = e(g,g)^y, \{T_i = g^{t_i}\}_{i \in [1,3n]})$.

KeyGen(PK, MK, S): Sun et al. consider the access structure supporting *don't care* condition in their scheme. The attribute set S in the input denotes for the positive attributes. The algorithm first chooses n randomness $r_1, \dots, r_n \in \mathbb{Z}_p$ and compute $r = \sum_{i \in [1,n]} r_i$. Then it computes $\hat{K} = g^{y-r}$ and $\left\{K_i = \begin{Bmatrix} g^{r_i/t_i}, & i \in S \\ g^{r_i/t_{n+i}}, & i \in \mathcal{U} \backslash S \end{Bmatrix}\right.$. Next, for $i \in [1, n]$, it computes $F_i = g^{r_i/t_{2n+i}}$. Finally, the secret key for S is $(\hat{K}, \{K_i, F_i\}_{i \in [1,n]})$.

Encrypt(PK, ω, W): In Sun *et al.*'s scheme, they only considered how to generate the encrypted index for a file, and the encryption of the file can be done using standard data encryption method. The algorithm first chooses a randomness $s \in \mathbb{Z}_p$ and computes $\hat{D} = g^s, \tilde{D} = Y^s$. Then it computes $\left\{D_i = \begin{Bmatrix} T_i^s, & i \in W \\ T_{n+i}^s, & i \in \mathcal{U} \backslash W \end{Bmatrix}\right.$. To simplify the case, here we assume that there is no don't care condition in W. Besides we omit the computation for the keyword indices since the ciphertext size is already proportional to the size of W.

Decrypt: They did not provide this algorithm in their paper since the file encryption is not in their consideration.:

B. Li et al.'s Scheme [38]

In Li *et al.*'s scheme, a part of the secret key will be outsourced to a semi-trusted party in order to reduce the decryption cost on the user ends. Therefore, there are two algorithms for generating secret keys, $\mathbf{KeyGen}_{out}$ and $\mathbf{KeyGen}_{in}$. The former generates the outsourced secret keys and the later generates the secret key for user ends. Besides, some necessary steps for generating secret keys will be performed in another algorithm called $\mathbf{KeyGen}_{init}$. In this paper, we use the algorithm $\mathbf{Decrypt}_{out}$ to denote the process of outsourcing decryption.

The access structure supported in [38] is a (d,t)-threshold-gate, where d is the threshold value which can be decided later in the $\mathbf{KeyGen}_{out}$ algorithm. By setting $d=t$, a threshold-gate is equivalent to an AND-gate.

$\mathbf{Setup}(1^\lambda)$: Let n be the size of the attribute universe. Taking the security parameter as input, the algorithm first chooses x and computes $g_1=g^x$. Then it randomly chooses $g_2,h,h_1,h_2,\dots,h_n\in\mathbb{G}$. Also, two cryptographic hash functions H_1,H_2 are chosen such that $H_1:\{0,1\}^*\to\mathbb{G}$ and $H_2:\mathbb{G}_T\to\{0,1\}^{\log p}$. The system parameter $PK=(g_1,g_2,h,h_1,h_2,\dots,h_n,H_1,H_2)$ and the master secret key $MK=x$.

$\mathbf{KeyGen}_{init}(MK)$: Taking as input the master secret key, the algorithm chooses $x_1\in\mathbb{Z}_p$ and computes $x_2=x-x_1 \bmod p$. The algorithm outputs $(OK_{KGCSP},OK_{TA})=(x_1,x_2)$.

$\mathbf{KeyGen}_{out}(A,\ OK_{KGCSP})$: Taking as input the access policy A and $OK_{KGCSP}=x_1$, the algorithm first chooses an $(d-1)$ degree polynomial $q(x)$ with $q(0)=x_1$. For $i\in A$, it then chooses $r_i\in\mathbb{Z}_p$ randomly, and computes $(d_{i0},d_{i1})=\left(g_2^{q(i)}(g_1h_i)^{r_i},g^{r_i}\right)$. Finally, the outsourcing secret key $SK_{KGCSP}=\{(d_{i0},d_{i1})\}_{i\in A}$.

$\mathbf{KeyGen}_{in}(OK_{TA},SK_{KGCSP})$: The attribute authority TA first chooses $r_\theta\in\mathbb{Z}_p$ and computes $SK_{TA}=(d_{\theta0},d_{\theta1})=\left(g_2^{x_2}(g_1h)^{r_\theta},g^{r_\theta}\right)$. The full secret key SK is set to be (SK_{KGCSP},SK_{TA}).

$\mathbf{Encrypt}(PK,W,M)$: The algorithms first chooses $s\in\mathbb{Z}_p$ and computes $C_0=M\cdot e(g_1,g_2)^s, C_1=g^s, C_\theta=(g_1h)^s, C_i=(g_1h_i)^s$, for $i\in W$. The ciphertext is $CT=(C_0,C_1,C_\theta,\{C_i\}_{i\in W})$.

$\mathbf{Decrypt}_{out}(PK,SK_{KGCSP},CT)$: The condition for successful decryption is that $|A\cap W|\geq d$. Assume that the condition holds, then there must be an authorized attribute set S where $|S|\geq d$. The algorithm first computes the Lagrange coefficients $\Delta_{i,S}(0)$ for $i\in S$. Then it computes

$$Q_{CT}=\frac{\prod_{i\in S}e(C_1,d_{i0})^{\Delta_{i,S}(0)}}{\prod_{i\in S}e(C_i,d_{i1})^{\Delta_{i,S}(0)}}=e(g,g_2)^{sx_1}.$$

Finally, Q_{CT} is outputted as the outsourcing decryption result.

$\mathbf{Decrypt}(PK,CT,Q_{CT},SK_{TA})$: An end user is able to recover the message M by computing

$$M=\frac{C_0\cdot e(C_\theta,d_{\theta1})}{Q_{CT}\cdot e(C_1,d_{\theta0})}.$$

C. Wang et al.'s Scheme [39]

In Wang *et al.*'s scheme, there is an entity called CS to perform the outsourcing decryption. After the attribute authority AA generating the secret key SK, the secret key will be divided into two parts (SK_1,SK_2). SK_2 will be sent to CS for outsourcing decryption, and SK_1 will be kept by the user. For decrypting a ciphertext, CS will first perform the algorithm **PreDecrypt** to generate the partial result.

$\mathbf{Setup}(1^\lambda)$: Taking as input the security parameter, the algorithm first chooses three cryptographic hash functions $H:\{0,1\}^*\to\mathbb{G}, H_1:\{0,1\}^*\to\mathbb{G}, H_2:\mathbb{G}_T\to\{0,1\}^{\log p}$. Then it randomly chooses $a,\alpha,\bar{\alpha}\in\mathbb{Z}_p$ and $v_i\in\mathbb{Z}_p$ for $i\in\mathcal{U}$, where $\mathcal{U}$ is the attribute universe. Finally, the algorithm outputs the system parameter

$$PK=(g^a,e(g,g)^\alpha,g^{\bar{\alpha}},\{PK_i=g^{v_i}\}_{i\in\mathcal{U}},H,H_1,H_2)$$

and the master secret key $MK=(\alpha,\bar{\alpha},\{v_i\}_{i\in\mathcal{U}})$.

$\mathbf{KeyGen}(MK,S_{uid})$: The attribute authority AA generates the secret key for S_{uid} as follows, where uid is the identity of the user. It first chooses $\alpha_1,\alpha_2\in\mathbb{Z}_p$ such that $\alpha_1+\alpha_2=\alpha \bmod p$. Also it randomly chooses $t,\delta_{uid}\in\mathbb{Z}_p$. Then it computes the secret key$SK=(SK_1,SK_2)$ where

$$SK_1=\left(\tilde{\alpha}=\frac{\bar{\alpha}}{\delta_{uid}},\ K=g^{\alpha_1}g^{at}\right),$$

$$SK_2=\left(\delta_{uid},E=g^{\alpha_2},L=g^t,\left\{K_i=H(i)^{\frac{t}{v_i}}\right\}_{i\in S_{uid}}\right).$$

Finally, SK_1 is sent to the user uid and (uid,S_{uid},SK_2) is sent to CS.

$\mathbf{Encrypt}(PK,(\mathcal{M},\rho),M)$: Wang *et al.*' adopts linear secret sharing scheme in their scheme. For a monotonic access structure, there is an efficient algorithm to transform it into a matrix $\mathcal{M}\in\mathbb{Z}_p^{\ell\times n}$ and a labelling function ρ, where ℓ and n are the parameter regarding to the access structure. We refer readers to [39] for further details. To encrypt a message M, the encryptor first chooses $s,y_2,\dots,\ y_n\in\mathbb{Z}_p$ and set a vector $\mathbf{v}=(s,y_2,\dots,\ y_n)$. Next, for $i=1$ to ℓ, it computes $\lambda_i=\mathcal{M}_i\cdot\mathbf{v}$, where $\mathcal{M}_i$ is the i-th row of $\mathcal{M}$. Then, the encryptor chooses $r_1,\dots,r_\ell\in\mathbb{Z}_p$ and computes

$$CT=\left(\begin{array}{c}C=M\cdot e(g,g)^{\alpha s},C'=g^s,\\ \{C_i=g^{a\lambda_i}H(\rho(i))^{r_i},D_i=\left(PK_{\rho(i)}\right)^{r_i}\}_{i\in[1,\ell]}\end{array}\right).$$

$\mathbf{PreDecrypt}(CT,SK_2)$: Let $I\subset\{1,\ 2,\dots,\ell\}$ be defined as $I=\{i:\rho(i)\in S_{uid}\}$. If S_{uid} satisfies the access structure of CT, then there is an efficient algorithm to compute a set of constants $\{\omega_i\in\mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I}\omega_i\lambda_i=s$. After obtaining $\{\omega_i\in\mathbb{Z}_p\}_{i\in I}$, CS is able to compute the partial result

$$A=\frac{\prod_{i\in I}e(C_i,L)^{\omega_i}}{e(C',E)\prod_{i\in I}e\left(D_i,K_{\rho(i)}\right)^{\omega_i}}.$$

Finally, CS outputs $CT'=(C,C',A)$ to the user uid.

$\mathbf{Decrypt}(CT',SK_1)$: The user uid is able to recover the message M by computing

$$M=\frac{C\cdot A}{e(C',K)}.$$

IV. Review on Wang *et al.*'s Scheme

In this section, we review the scheme which Wang *et al.* [2] proposed as follows:

Wang *et al.* exploit a series of AND gates on multi-value attributes as the access structure. Assume that the total number of attributes is n, and they are indexed as

$$U = \{att_1, att_2, \dots, att_n\}.$$

For each attribute

$$att_i \in U, (i = 1,2, \dots, n),$$

let

$$V_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,n_i}\}$$

be a set of possible values of att_i. Then the attribute list S for a user is

$$S = (x_1, x_2, \dots, x_n),$$

where $x_i \in V_i$. The access policy in a ciphertext is

$$W = (W_1, W_2, \dots, W_n),$$

where $W_i \in V_i$. Let **PairGen** be an algorithm that, on inputting a security parameter 1^λ, outputting a tuple

$$\Upsilon = (p, \mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_2, e_0, e_1),$$

where $\mathbb{G}_0$, $\mathbb{G}_1$ and $\mathbb{G}_2$ have the same prime order p, and

$$e_i\colon \mathbb{G}_0 \times \mathbb{G}_i \to \mathbb{G}_{i+1}, i = \{0,1\}$$

are efficient non-degenerate multilinear maps such that

$$\forall\ \alpha, \beta \in \mathbb{Z}_p,$$

$$e_i(g_0^\alpha, g_i^\beta) = e_i(g_0, g_i)^{\alpha\beta}.$$

Wang *et al.*'s scheme consists of six algorithms, including **Setup**, **KeyGen**, **Encrypt**, **Trapdoor**, **Test**, and **Decrypt**. We give the details for these algorithms in the follows.

- **Setup** $(1^\lambda) \to (PK, MSK)$

The algorithm runs the generator algorithm **PairGen**(1^λ) and gets the group and the multilinear mapping description

$$\Upsilon = (p, \mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_2, e_0, e_1),$$

where $\mathbb{G}_0$ is generated by g_0. The algorithm randomly chooses $\alpha, \beta \in \mathbb{Z}_p$ and a hash function $H\colon \{0,1\}^* \to \mathbb{G}_0$. Let

$$A = e_0(g_0, g_0)^\alpha,$$

$$B = g_0^\beta.$$

The system public key is

$$PK = (\Upsilon, g_0, A, B, H)$$

and the master secret key is $MSK = (\alpha, \beta)$.

- **KeyGen** $(MSK, S) \to SK$

The key generation algorithm will take as input a set of attributes

$$S = (x_1, x_2, \dots, x_n)$$

and output the secret key SK. The key generation algorithm selects random

$$r_i \in \mathbb{Z}_p (i = 1,2, \dots, n)$$

and sets

$$r = \sum_{i=1}^{n} r_i$$

and computes

$$\widetilde{D} = g_0^{(\alpha+r)/\beta}.$$

Randomly select $r' \in \mathbb{Z}_p$, for each attribute $x_i \in S$, compute

$$D_i = {g_0}^{r_i} \cdot H(x_i)^{r'},$$

and set

$$\widehat{D} = g_0^{r'}.$$

$$\breve{D} = \prod_{i=1}^{n} H(x_i)^\beta.$$

The secret key is defined as

$$SK = \left(\widetilde{D}, \breve{D}, \widehat{D}, \{D_i\} \middle| (x_i \in S)\right).$$

- **Encrypt** $(1^\lambda, M \in \mathbb{G}_1, PK, \omega, W) \to (CT, I_\omega)$

Give a message $M \in \mathbb{G}_1$ and an AND gate

$$W = (W_1, W_2, \dots, W_n),$$

and the corresponding keyword ω, the encryption algorithm selects random numbers $s, s' \in \mathbb{Z}_p^*$ and sets

$$C = M \cdot A^s,$$

$$\tilde{C} = B^s,$$

$$\hat{C} = g_0^s.$$

For each attribute W_i in the AND gate W, compute

$$C_i = H(W_i)^s \text{ for } i = 1 \text{ to } n,$$

and set

$$\check{C}_1 = e_0(B, g_0^{\omega s'}).$$

$$\check{C}_2 = g_0^{ss'}.$$

The ciphertext is

$$CT = (C, \tilde{C}, \hat{C}, \{C_i\}_{i=1}^{n}),$$

and index

$$I_\omega = (C_i, \check{C}_1, \check{C}_2).$$

- **Trapdoor** $(SK, \omega) \to t_\omega$

User U_i generates the trapdoor of his chosen keyword ω as:

$$t_\omega = e_0(\breve{D}, g_0^\omega).$$

- **Test** $(I_\omega, t_\omega) \to 0$ or 1

Given the trapdoor t_ω and index I_ω, the cloud server checks the following formula

$$e_1(t_\omega, \check{C}_2) = e_1(\prod_{i=1}^{n} C_i, \check{C}_1).$$

If the formula holds, return 1 and 0 otherwise.

- **Decrypt** $(I_\omega, t_\omega) \to M$

If the attribute list S satisfies the access policy W i.e. $x_i =$

W_i, $(i = 1,2,\dots,n)$. On inputting ciphertext CT and secret key SK, output a message M as:

$$E = \prod_{i=1}^{n} \frac{e_0(D_i, \hat{C})}{e_0(\widehat{D}, C_i)}$$
$$= e_0(g_0, g_0)^{rs}$$
$$M = C/(e_0(\widetilde{D}, \tilde{C})/E).$$

V. Comments on Wang *et al.*'s Searchable Encryption

First, the length of a ciphertext is obviously not independent of the number of attributes. In their scheme,

$$C_i = H(W_i)^s \ (i = 1,2,\dots,n)$$

need to be published in a ciphertext

$$CT = (C, \tilde{C}, \hat{C}, C_i)$$

and an index

$$I_\omega = (C_i, \check{C}_1, \check{C}_2).$$

When it comes to "constant-size ciphertext" [36], [37], we usually mean "the length of a ciphertext is independent of the number of attributes in the *access* structure".

Another problem is that they did not achieve the hidden policy. To guess if a keyword w^* has been used in the **Encrypt** algorithm, an attacker verifies whether the following formula holds:

$$e_1(\hat{C}, \check{C}_1) = e_1(\check{C}_2, e_0(B, g_0^{w^*}))$$

The correctness analysis is given as follows.

$$e_1(\hat{C}, \check{C}_1) = e_1\left(g_0^s, e_0(B, g_0^{ws'})\right)$$
$$= e_1(g_0^s, e_0(B, g_0^w)^{s'})$$
$$= e_1\left(g_0^{ss'}, e_0(B, g_0^w)\right)$$
$$= e_1(\check{C}_2, e_0(B, g_0^w))$$

The formula holds due to the linearity of multi-linear maps.

Furthermore, to test if an attribute value W_i^* has been used in the **Encrypt** algorithm, an attacker checks whether the following formula hold:

$$e_0(C_i, g_0) = e_0(H(W_i^*), \hat{C})$$

It is because that

$$e_0(C_i, g_0) = e_0(H(W_i)^s, g_0)$$
$$= e_0(H(W_i), g_0^s)$$
$$= e_0(H(W_i), \hat{C}).$$

The attacker only needs to perform $O(n \times \max\{v_{i,n_i}\})$ pairings to guess the n attribute values in a ciphertext.

VI. An Improvement on Wang *et al.*'s Scheme

One solution is to add a new ciphertext component:

$$\bar{C} = \prod_{i=1}^{n} C_i$$

instead of directly publishing all C_i's such that the ciphertext

$$CT = (C, \tilde{C}, \hat{C}, \bar{C})$$

and the index

$$I_\omega = (\bar{C}, \check{C}_1, \check{C}_2).$$

In this setting an attacker needs to guess the n values corresponding to the n attributes, respectively, at one time, and thus the number of all possible combinations becomes exponentially large, i.e.

$$(v_{1,n} \times v_{2,n} \times \dots \times v_{n,n}) \geq 2^n.$$

Furthermore, the length of the ciphertext now is indeed independent of the number of the attributes in the access structure.

Besides, we can add a new component

$$\overline{D} = \prod_{i=1}^{n} D_i$$
$$= g_0^r \left(\prod_{i=1}^{n} H(x_i)\right)^{r'}$$

instead of using all D_i's such that the secret key is $(\widetilde{D}, \check{D}, \widehat{D}, \overline{D})$. Thus, the size of the secret key can also be independent of the number of the attributes.

Since we have changed the forms of ciphertexts and secret keys, the **Test** and the **Decryp**t algorithms need to be changed accordingly. The **Test** algorithm is changed to

$$e_1(t_\omega, \check{C}_2) = e_1(\bar{C}, \check{C}_1).$$

To compute $e_0(g_0, g_0)^{rs}$ in the **Decrypt** algorithm, one now computes

$$E = \frac{e_0(\overline{D}, \hat{C})}{e_0(\widehat{D}, \bar{C})}$$
$$= e_0(g_0, g_0)^{rs}.$$

Note that the number of pairings is now independent of the number of the attributes in the access structure.

VII. Comparison

In this section, we compare our improved scheme with the schemes of [27], [38], [39] in terms of ciphertext size, secret key size, and the decryption cost.

A. Sun et al.'s Scheme [27]

In Sun *et al.*'s scheme [27], a secret key is $(\widehat{K}, \{K_i, F_i\}_{i\in[1,n]})$, where

$$\widehat{K} = g^{y-r}, K_i = \begin{cases} g^{r_i/t_i}, & i \in S \\ g^{r_i/t_{n+i}}, & i \in \mathcal{U}\backslash S \end{cases}, F_i = g^{r_i/t_{2n+i}},$$

and n is the size of the attribute universe. Therefore, the secret key size is $2n + 1$ elements in $\mathbb{G}$.

A ciphertext in their scheme is

$$\left(\widehat{D} = g^s, \widetilde{D} = Y^s, \left\{D_i = \begin{cases} T_i^s, & i \in W \\ T_{n+i}^s, & i \in \mathcal{U}\backslash W \end{cases}\right\}\right).$$

Thus the ciphertext size is $n+1$ elements in $\mathbb{G}$ plus one element in $\mathbb{G}_T$. Since Sun *et al*. only consider how to generate the encrypted indices, there is no **Decrypt** algorithm in their paper.

B. Li et al.'s Scheme [38]

Li *et al*.'s scheme supports threshold-gate in the access structure. Let d be the threshold value for the access policy. The secret key of a user is with the form

$$SK = (SK_{KGCSP}, SK_{TA}),$$

where

$$SK_{KGCSP} = \{(d_{i0}, d_{i1})\}_{i\in A} = \left\{\left(g_2^{q(i)}(g_1 h_i)^{r_i}, g^{r_i}\right)\right\}_{i\in A}$$

and

$$SK_{TA} = (d_{\theta 0}, d_{\theta 1}) = \left(g_2^{x_2}(g_1 h)^{r_\theta}, g^{r_\theta}\right).$$

Though the size of SK_{TA} is independent of the number of attributes in A, the size of SK_{KGCSP} is $2|A|$.

Thus the size of a full secret key is $2|A|+2$ elements in $\mathbb{G}$, where $|A|$ denotes the size of A. A ciphertext in [38] is with the form

$$\begin{pmatrix} C_0 = M\cdot e(g_1,g_2)^s, C_1 = g^s, \\ C_\theta = (g_1 h)^s, \{C_i = (g_1 h_i)^s\}_{i\in W}\end{pmatrix}.$$

Thus the size of a ciphertext is $|W|+2$ elements in $\mathbb{G}$ plus one element in $\mathbb{G}_T$, where $|W|$ is the size of the attribute set W.

The decryption process in [38] is twofold. One is the cost of $\mathbf{Decrypt}_{out}$, which is performed by a semi-trusted server, another is the cost of **Decrypt**, which is performed by the user. In algorithm $\mathbf{Decrypt}_{out}$, the server needs to compute

$$Q_{CT} = \frac{\prod_{i\in S} e(C_1, d_{i0})^{\Delta_{i,S}(0)}}{\prod_{i\in S} e(C_i, d_{i1})^{\Delta_{i,S}(0)}},$$

where $|S| = |A\cap W| \geq d$, and thus at least $2d$ bilinear maps are necessary. In the user side, a user needs to compute

$$\frac{C_0\cdot e(C_\theta, d_{\theta 1})}{Q_{CT}\cdot e(C_1, d_{\theta 0})}$$

to recover the message M. Therefore, two pairings are necessary for a user to recover M. Totally, to decrypt a ciphertext needs at least $2d+2$ pairings.

C. Wang et al.'s Scheme [39]

In Wang *et al*.'s scheme, a secret key SK consists of the following two parts,

$$SK_1 = \left(\tilde{\alpha} = \frac{\bar{\alpha}}{\delta_{uid}},\ K = g^{\alpha_1} g^{at}\right),$$

$$SK_2 = \left(\delta_{uid}, E = g^{\alpha_2}, L = g^t, \left\{K_i = H(i)^{\frac{t}{v_i}}\right\}_{i\in S_{uid}}\right).$$

As the same case as that in [38], the size of SK_1 is independent of the number of the user's attributes, which is an element in $\mathbb{Z}_p$ plus an element in $\mathbb{G}$, while the size of SK_2 is proportional to the number of the user's attributes, which is an element in $\mathbb{Z}_p$ plus $|S_{uid}|+2$ elements in $\mathbb{G}$.

A ciphertext in [39] consists of the following components

$$CT = \begin{pmatrix} C = M\cdot e(g,g)^{\alpha s}, C' = g^s, \\ \left\{C_i = g^{a\lambda_i} H(\rho(i))^{r_i}, D_i = \left(PK_{\rho(i)}\right)^{r_i}\right\}_{i\in[1,\ell]}\end{pmatrix},$$

where ℓ is the number of rows in $\mathcal{M}$, which is equal to or greater than the number of attributes in the access structure. Therefore, the size of a ciphertext is at least $2|W|+1$ elements in $\mathbb{G}$ plus an element in $\mathbb{G}_T$, where $|W|$ denotes the number of attributes in the access structure.

The decryption cost is also twofold. Recall that in the algorithm **PreDecrypt**, CS needs to compute

$$A = \frac{\prod_{i\in I} e(C_i, L)^{\omega_i}}{e(C', E)\prod_{i\in I} e\left(D_i, K_{\rho(i)}\right)^{\omega_i}}.$$

Here I is defined as $I = \{i: \rho(i)\in S_{uid}\}$, thus $|I| \leq |S_{uid}|$. We have that the number of pairings needed for CS is

$$2|I|+1 \leq 2|S_{uid}|+1.$$

On the user side, a user needs to compute

$$\frac{C\cdot A}{e(C', K)}$$

to recover the message M, thus a bilinear map is necessary. Totally, to decrypt a ciphertext needs approximately

D. Our Improved Scheme

In our improved scheme, a ciphertext and a secret key are with the form $CT = (C, \tilde{C}, \hat{C}, \bar{C})$ and $SK = \left(\tilde{D}, \breve{D}, \hat{D}, \bar{D}\right)$, respectively. Both the sizes are independent of the number of attributes.

To decrypt a ciphertext, a user first computes

$$E = \frac{e_0\left(\bar{D}, \hat{C}\right)}{e_0\left(\hat{D}, \bar{C}\right)},$$

then recover the message by computing

$$M = \frac{C}{\frac{e_0\left(\tilde{D}, \tilde{C}\right)}{E}}.$$

Thus, three e_0 operations is needed.

The notations used in the comparison is shown in Table 1. To simplify the case for comparison, we have to make some assumptions. Here we consider only AND-gate access structure. For [38], we assume that $|S| = |A\cap W| = d$. For [39], we assume that $|I| = |S_{uid}| = |A|$. The comparison is summarized in Table II.

TABLE I: NOTATIONS

T_e	The computation time of a pairing
T_{e_0}	The computation time of e_0
$\lvert\mathbb{Z}_p\rvert$	The size of an element in $\mathbb{Z}_p$
$\lvert\mathbb{G}\rvert$	The size of an element in $\mathbb{G}$
$\lvert\mathbb{G}_T\rvert$	The size of an element in $\mathbb{G}_T$
$\lvert\mathbb{G}_0\rvert$	The size of an element in $\mathbb{G}_0$
$\lvert\mathbb{G}_1\rvert$	The size of an element in $\mathbb{G}_1$
n	The size of the attribute universe
A	The set of attributes corresponding to a secret key
W	The set of attributes corresponding to a ciphertext
d	The threshold value used in [38]

VIII. CONCLUSION

To protect the privacy, users usually encrypt their files before uploading to clouds. Searchable encryption has then been proposed for efficient searches over encrypted files on clouds. In this paper, we first give a review on Wang *et al*.'s ABKS scheme, and then give a cryptanalysis to their work. Our analysis shows that Wang *et al*.'s scheme did not achieve hidden policy, and the ciphertext lengths of their scheme are not independent of the number of attributes. We further proposed an improvement to solve the aforementioned problems. By applying our method, the improved scheme achieves constant-size ciphertext. In the future, we will focus on proving the security of our improved scheme under standard security models.

TABLE II: COMPARISON

	Secret Key Size	Ciphertext Size	Decryption Cost
[27]	$(2n+1)\lvert\mathbb{G}\rvert + (n+1)\lvert\mathbb{G}_T\rvert$	$(n+1)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	---
[38]	$(2\lvert A\rvert+2)\lvert\mathbb{G}\rvert$	$(\lvert W\rvert+2)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$(2d+2)T_e$
[39]	$(\lvert A\rvert+3)\lvert\mathbb{G}\rvert + 2\lvert\mathbb{Z}_p\rvert$	$(2\lvert W\rvert+1)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$(2\lvert A\rvert+1)T_e$
Ours	$4\lvert\mathbb{G}_0\rvert$	$2\lvert\mathbb{G}_0\rvert + 2\lvert\mathbb{G}_1\rvert$	$3T_{e_0}$

ACKNOWLEDGMENT

This work was partially supported by Taiwan Information Security Center at National Sun Yat-sen University (TWISC@NSYSU) and the Ministry of Science and Technology of Taiwan under grant MOST 107-2218-E-110-014. It also was financially supported by the Information Security Research Center at National Sun Yat-sen University in Taiwan and the Intelligent Electronic Commerce Research Center from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

REFERENCES

[1] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. 2000 IEEE Symposium on Security and Privacy, 2000*, pp. 44-55.

[2] H. Wang, X. Dong, and Z. Cao, "Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search," *IEEE Transactions on Services Computing*, 2017.

[3] M. H. Ameri, M. Delavar, J. Mohajeri, and M. Salmasizadeh, "A key-policy attribute-based temporary keyword search scheme for secure cloud storage," *IEEE Transactions on Cloud Computing*, 2018.

[4] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, "Attributebased keyword search over hierarchical data in cloud computing," *IEEE Transactions on Services Computing*, 2018.

[5] Y. Miao, J. Ma, X. Liu, J. Weng, and H. Li, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Transactions on Services Computing*, 2018.

[6] H. Wang, X. Dong, and Z. Cao, "Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search," *IEEE Transactions on Services Computing*, 2018.

[7] D. Boneh and A. Silverberg, *Applications of Multilinear Forms to Cryptography*.

[8] W. Sun, B. Wang, N. Cao, M. Li *et al*., "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 3025-3035, 2014.

[9] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 127-138, 2015.

[10] K. Liang and W. Susilo, 2015, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1981-1992.

[11] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, May 2010, pp. 62-91.

[12] K. Yang and X. Jia, "Attributed-based access control for multi-authority systems in cloud storage," in *Proc. 2012 IEEE 32nd International Conference on Distributed Computing Systems (ICDCS)*, June 2012, pp. 536-545.

[13] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, May 2005, pp. 457-473.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, October 2006, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. the 13th ACM Conference on Computer and Communications Security*, pp. 89-98.

[15] J. Han, W. Susilo, Y. Mu, andJ. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150-2162, 2012.

[16] W. Liu, J. Liu, Q. Wu, B. Qin, and Y. Zhou, "Practical direct chosen ciphertext secure key-policy attribute-based encryption with public ciphertext test," in *Proc. European Symposium on Research in Computer Security*, Springer, Cham, pp. 91-108, September 2014.

[17] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symposium on Security and Privacy*, May 2007, pp. 321-334.

[18] Y. Zhang, A. Wu, and D. Zheng, "Efficient and privacy-aware attribute-based data sharing in mobile cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1039-1048, 2018.

[19] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42-61, 2017.

[20] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1-12, 2018.

[21] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. the 14th ACM Conference on Computer and Communications Security*, October 2007, pp. 456-465, ACM.

[22] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 261-270, ACM, April 2010.

[23] T. Jung, X. Y. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 190-199, 2015.

[24] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384-1393, 2015.

[25] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 126-138, 2015.

[26] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, May 2004, pp. 506-522.

[27] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in *Proc. INFOCOM*, April 2014, pp. 226-234.

[28] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proc. Infocom*, April 2014, pp. 522-530.

[29] Q. Zheng, X. Li, and A. Azgin, "CLKS: Certificateless keyword search on encrypted data," in *Proc. International Conference on Network and System Security*, Springer, Cham, November 2015, pp. 239-253.

[30] H. Li, D. Liu, K. Jia, and X. Lin, "Achieving authorized and ranked multi-keyword search over encrypted cloud data," in *Proc. 2015 IEEE International Conference on Communications*, June 2015, pp. 7450-7455.

[31] C. C. Lee, C. T. Li, C. L. Chen, and S. T. Chiu, "A searchable hierarchical conditional proxy re-encryption scheme for cloud storage services," *Information Technology and Control*, vol. 45, no. 3, pp. 289-299, 2016.

[32] J. Li, Y. Shi, and Y. Zhang, “Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage,” *International Journal of Communication Systems*, vol. 30, no. 1, p. e2942, 2017.
[33] Q. Wang, Y. Zhu, and X. Luo, “Multi-user searchable encryption with fine-grained access control without key sharing,” in *Proc. 2014 3rd International Conference on Advanced Computer Science Applications and Technologies*, December 2014, pp. 145-150.
[34] Y. Miao, J. Ma, X. Liu, F. Wei, Z. Liu, and X. A. Wang, 2016, “m2-ABKS: Attribute-based multi-keyword search over encrypted personal health records in multi-owner setting,” *Journal of Medical Systems*, vol. 40, no. 11, p. 246.
[35] J. Lai, X. Zhou, R. Deng, X., Li, and K. Chen, “Expressive search on encrypted data,” in *Proc. the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, 2013, pp. 243-252.
[36] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. De Panafieu, and C. Ràfols, “Attribute-based encryption schemes with constant-size ciphertexts,” *Theoretical Computer Science*, vol. 422, pp. 15-38, 2012.
[37] C. Chen, Z. Zhang, and D. Feng, “Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost,” in *Proc. International Conference on Provable Security*, Berlin, Heidelberg: Springer, 2011, pp. 84-101.
[38] J. Li, X. Lin, Y. Zhang, and J. Han, “KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage,” *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2016.
[39] S. Wang, D. Zhang, Y. Zhang, and L. Liu, “Efficiently revocable and searchable attribute-based encryption scheme for mobile cloud storage,” *IEEE Access*, vol. 6, pp. 30444–30457, 2018.

Yi-Fan Tseng was born in Kaohsiung, Taiwan. He received the Ph.D. degree and MS degree in computer science and engineering from National Sun Yat-sen University, Taiwan, in 2014 and 2018, respectively. From 2018 to 2019, as a postdoctoral researcher, he joined the research group of Taiwan Information Security Center at National Sun Yat-sen University (TWISC@NSYSU). In 2019, he has joined the Department of Computer Science, National Chengchi University, Taipei, Taiwan. His research interests include cloud computing and security, network and communication security, information security, cryptographic protocols, and applied cryptography.

Chun-I Fan received the M.S. degree in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1998. From 1999 to 2003, he was an associate researcher and a project leader with Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taoyuan, Taiwan. In 2003, he joined the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. He has been a full professor since 2010 and a distinguished professor since 2019. His current research interests include applied cryptology, cryptographic protocols, and information and communication security.

Prof. Fan is the chairman of Chinese Cryptology and Information Security Association (CCISA), Taiwan, and the chief executive officer (CEO) of Telecom Technology Center (TTC), Taiwan. He also is the director of Information Security Research Center and was the CEO of "Aim for the Top University Plan" Office at National Sun Yat-sen University. He was the recipient of the Best Student Paper Awards from the National Conference on Information Security in 1998, the Dragon Ph.D. Thesis Award from Acer Foundation, and the Best Ph.D. Thesis Award from the Institute of Information and Computing Machinery in 1999. Prof. Fan won the Engineering Professors Award from Chinese Institute of Engineers - Kaohsiung Chapter in 2016 and he is also an outstanding faculty in Academic Research in National Sun Yat-sen University.

Si-Jing Wu is now an MS student in National Sun Yat-sen University, Kaohsiung, Taiwan. Her major is in computer science and engineering. Her research interests include cloud computing and cloud storage, network and communication security, and applied cryptography.

Hsin-Nan Kuo was born in Pingtung, Taiwan. Presently, he is a Ph.D. fellow in the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. His current research includes cloud security, network/information security, and applied cryptography.

Jheng-Jia Huang was born in Kaohsiung, Taiwan. He received the M.S. degree in information management from National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, in 2012. Also, he received the Ph.D. degree in computer science and engineering from the National Sun Yat-sen University, Kaohsiung, Taiwan, in 2019. In the same year, he has been a CEO special assistant of the Telecom Technology Center, Kaohsiung, Taiwan. His current research interests include cloud computing and security, social network security and authentication, network and communication security, information security, and applied cryptography.

Dr. Huang is the deputy secretary general of the Chinese Cryptology and Information Security Association and the Quality Control/ Quality Assurance Supervisor of the Telecom Technology Center. He was the recipient of the Phi Tau Phi award from the National Sun Yat-sen University in 2019.