Comments on "Verifiable and Exculpable Outsourced Attribute-Based Encryption for Access Control in Cloud Computing"

Hu Xiong and Jianfei Sun

Abstract—Recently in IEEE Transactions on Dependable and Secure Computing (TDSC) (doi: 10.1109/TDSC.2015.2499755), Ma et al. proposed a new construction of attribute-based encryption (ABE) which can outsource the complicated encryption task to Encryption Service Provider (ESP) in a verifiable manner. Despite the authors claimed that the results of the outsourced encryption can be checked by the user, we show that Ma et al.'s proposal fails to provide the verifiability property for outsourced encryption, the most essential security goal that a verifiable computation scheme should achieve. Specifically, by giving concrete attacks, we demonstrate that the ESP can return forged intermediate ciphertext to the user without being detected.

Index Terms—Attribute-based encryption, outsourced computation, verifiability

1 Introduction

To realize flexible one-to-many encryption, attribute-based encryption (ABE) has been suggested as the extension of identity-based encryption such that access policies and ascribed attributes are embedded into the private keys and ciphertexts respectively. The ABE ciphertext can only be decrypted by a private key in case the corresponding attributes satisfy the pre-defined access policy. Featured with its expressiveness, ABE is viewed as a promising solution to achieve the flexible access control. By considering the prohibitively high encryption and decryption cost of ABE primitive, Ma et al. [1] proposed a new construction of ABE scheme to provide both verifiable outsourced encryption and decryption recently. In addition to enable the verifiable decryption for the receiver, the sender can also offload intensive encryption task to the untrusted Encryption Service Provider (ESP) by demanding the ESP to generate the intermediate ciphertext for the sender. Moreover, the intermediate ciphertext returned from the ESP can be verified and then combined into the final ciphertext by the sender. Despite the authors claimed that a sender could check the correctness of the intermediate ciphertext returned from ESP by using a batch verification algorithm, unfortunately, after carefully observing the scheme in [1], we demonstrate that this scheme fails to offer the verifiability to the outsourced encryption. To be more specific, the ESP can generate the forged intermediate ciphertex without being detected by the batch verification algorithm.

2 Review of the Ma et al.'s Scheme

We only review the *Setup*, *Encrypt.out* and *Verify.enc* algorithms of the scheme in [1], and readers may refer to [1] for the details of *KeyGen*, *KeyGen.out*, *Encrypt.user*, *Decrypt.out* and *Decrypt.user* algorithms. Given a prime number p and two cyclic multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ featured with same order p, the mapping $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is said to be a bilinear map in case this map is equipped with the bilinearity and non-degeneracy. Concretely, $e(u^a, v^b) = e(u, v)^{ab}$ for $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_p$; and $e(g, g) \neq 1_{\mathbb{G}_T}$, where g represents a generator chosen randomly from $\mathbb{G}$.

Setup (λ, U). Given λ as the security parameter and U as the universe attributes respectively, this algorithm first selects a bilinear map $\mathsf{BM}=(\mathbb{G}, \mathbb{G}_T, e, p)$. It then randomly picks several generators $g, h, u, v, w \in \mathbb{G}$ at random and chooses a random $\alpha \in \mathbb{Z}_p$. It finally assigns $(\mathsf{BM}, g, h, u, v, w, e(g, g)^{\alpha})$ as the public parameter PK and α as the master secret key MSK, respectively.

Encrypt.out (PK, N). Given the public parameters $\mathsf{P}K$ and a number N representing the maximum bound of N arrows for any LSSS access structure, this algorithm is carried out by an untrusted ESP. It first chooses $s' \in \mathbb{Z}_p$ at random and calculates $C_0 = g^{s'}$. Then, it randomly picks $\lambda'_j, x'_j, t'_j \in \mathbb{Z}_p$ and computes $C_{j,1} = w^{\lambda'_j} v^{t'_j}$, $C_{j,2} = (u^{x'_j} h)^{-t'_j}, C_{j,3} = g^{t'_j}$ for j ranges from 1 to N. Lastly, $IT = (s', C_0, \{\lambda'_j, x'_j, t'_j, C_{j,1}, C_{j,2}, C_{j,3}\}_{j\in[1,N]})$ is returned to the requesting sender as the intermediate ciphertext.

Verify.enc(PK, IT). After receiving the public parameters PK and an intermediate ciphertext $IT = (s', C_0, \{\lambda'_j, x'_j, t'_j, C_{j,1} = w^{\lambda'_j} v^{t'_j}, C_{j,2} = (u^{x'_j} h)^{-t'_j}, C_{j,3} = g^{t'_j}\}_{j\in[1,N]})$, the algorithm is performed by the sender to check the correctness of the IT. It first picks b as the security parameter. Then it calculates $y'_j = x'_j \times (-t'_j) \bmod p$ and $F_j = C_{j,1} C_{j,2} C_{j,3}$ such that $j \in [1, N]$. Finally, it verifies the correctness of the intermediate ciphertext IT by checking whether $C_0 \stackrel{?}{=} g^{s'}$ and executes the algorithm *Batch-Verify* with the input $(\{F_j, \lambda'_j, t'_j, y'_j, -t'_j, t'_j\})$ such that $j \in [1, N]$ and $y'_j = -x'_j t'_j \bmod p$.

3 Analysis of the Ma et al.'s Scheme

Despite the scheme in [1] claimed that a sender could validate whether an untrusted ESP generates the intermediate ciphertext IT honestly or not, we demonstrate that the malicious ESP can intentionally forge an illegal intermediate ciphertext IT without being detected by the *Batch-Verify* algorithm. Specifically, the malicious ESP could trick an honest user to believe that the forged intermediate ciphertext was correctly computed using either properly chosen numbers or leveraging an older intermediate cipheretext. The details of both situations of the forgery attack on the batch verification algorithm are as follows:

Scenario 1: The ESP randomly chooses $s', \lambda'_j, x'_j, t'_j \in \mathbb{Z}_p$ and dishonestly calculates $C'_0 = g^{s'}, C'_{j,1} = w^{\lambda'_j}, C'_{j,2} = (u^{x'_j} h)^{-t'_j}$ and $C'_{j,3} = (gv)^{t'_j}$ for j ranges from 1 to N. Then, $IT' = (s', C'_0, \{\lambda'_j, x'_j, t'_j, C'_{j,1}, C'_{j,2}, C'_{j,3}\}_{j\in[1,N]})$ is not a valid IT. However, this intermediate ciphertext IT' can be batch verified successfully as follows. Given the public parameters $\mathsf{PK} = (\mathsf{BM}, g, h, u, v, w, e(g, g)^{\alpha})$ and IT', the user first checks whether $C'_0 \stackrel{?}{=} g^{s'}$ or not, and then computes $F'_j = C'_{j,1} C'_{j,2} C'_{j,3} = w^{\lambda'_j} u^{y'_j} h^{-t'_j} (gv)^{t'_j} = w^{\lambda'_j} v^{t'_j} u^{y'_j} h^{-t'_j} g^{t'_j}$ where $\forall j \in [1, N]$ and $y'_j = -x'_j t'_j$. Next, the user picks $a_1, \ldots, a_N \in \{0, 1\}^b$ at random, and computes $A = \sum_{j=1}^{N} \lambda'_j a_j \bmod p$, $B = \sum_{j=1}^{N} t'_j a_j \bmod p$, $C = \sum_{j=1}^{N} y'_j a_j \bmod p$, $D = -B$, $E = B$, and $F' = \prod_{j=1}^{N} F'^{a_j}_j$. It's obvious that $F' = w^A v^B u^C h^D g^E$.

Scenario 2: Given a valid intermediate ciphertext $IT = (s, C_0, \{\lambda_j, x_j, t_j, C_{j,1}, C_{j,2}, C_{j,3}\}_{j\in[1,N]})$, the ESP randomly chooses $\alpha, \beta, \gamma \in \mathbb{Z}_p$ and dishonestly calculates $C'_0 = C_0 \cdot g^{\gamma}, C'_{j,1} = C_{j,1} \cdot g^{-\alpha-\beta}, C'_{j,2} = C_{j,2} \cdot g^{\alpha}, C'_{j,3} = C_{j,3} \cdot g^{\beta}$ for $j \in [1, N]$. Then, $IT' = (s', C'_0, \{\lambda'_j, x'_j, t'_j, C'_{j,1}, C'_{j,2}, C'_{j,3}\}_{j\in[1,N]}) = (s + \gamma, C'_0, \{\lambda_j, x_j, t_j, C'_{j,1}, C'_{j,2}, C'_{j,3}\}_{j\in[1,N]})$ is not a valid IT. However, this intermediate ciphertext IT' can be batch verified successfully as follows. Given the public parameters $\mathsf{PK} = (\mathsf{BM}, g, h, u, v, w, e(g, g)^{\alpha})$ and $IT' = (s', C'_0, \{\lambda'_j,$

• *The authors are with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610051, China. E-mail: {xionghu.uestc, sjf215.uestc}@gmail.com.*

Manuscript received 10 Aug. 2016; revised 17 Dec. 2016; accepted 26 May 2017. Date of current version 7 July 2017.
(Corresponding author: Hu Xiong.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TDSC.2017.2710119

TABLE 1
Efficiency Comparison

Schemes	Enc (Server)
[1]	$(5N+1)$Exp
First attack	$(4N+1)$Exp
Second attack	4Exp

$x'_j, t'_j, C'_{j,1}, C'_{j,2}, C'_{j,3}\}_{j\in[1,N]}$), the user first checks whether $C'_0 \stackrel{?}{=} g^{s'}$ or not, and then computes $F'_j = C'_{j,1}C'_{j,2}C'_{j,3} = w^{\lambda'_j} \cdot v^{t'_j} \cdot g^{-\alpha-\beta} \cdot u^{y'_j} \cdot h^{-t'_j} \cdot g^{\alpha} \cdot g^{t'_j} \cdot g^{\beta} = w^{\lambda'_j} v^{t'_j} u^{y'_j} h^{-t'_j} g^{t'_j}$ where $\forall j \in [1, N]$ and $y'_j = -x'_j t'_j$. Next, the user picks $a_1, \ldots, a_N \in \{0,1\}^b$ at random, and computes $A = \sum_{j=1}^{N} \lambda'_j a_j \bmod p$, $B = \sum_{j=1}^{N} t'_j a_j \bmod p$, $C = \sum_{j=1}^{N} y'_j a_j \bmod p$, $D = -B$, $E = B$, and $F' = \prod_{j=1}^{N} F'^{a_j}_j$. It's obvious that $F' = w^A v^B u^C h^D g^E$.

Algorithm 1. Batch-Verify

Input: Given the public parameters $\mathsf{PK} = (\mathsf{BM}, g, h, u, v, w, e(g,g)^{\alpha})$ and a batch of $(F_1, \lambda_1, t_1, y_1, -t_1, t_1)$, $\ldots$, $(F_N, \lambda_N, t_N, y_N, -t_N, t_N)$ such that $\lambda_i, t_i, y_i, -t_i, t_i$ are chosen randomly from $\mathbb{Z}_p$, F_i is selected from $\mathbb{G}$ at random, and i ranges from 1 to N, as the input.

Output: Running the following steps to validate the correctness of each $F_i = w^{\lambda_i} v^{t_i} u^{y_i} h^{-t_i} g^{t_i}$ for i ranges from 1 to N.

1: Select $a_1, \ldots, a_N \in \{0,1\}^b$ randomly;
2: Calculate $A = \sum_{i=1}^{N} \lambda_i a_i \bmod p$, $B = \sum_{i=1}^{N} t_i a_i \bmod p$, $C = \sum_{i=1}^{N} y_i a_i \bmod p$, $D = -B$, $E = B$ and $F = \prod_{i=1}^{N} F_i^{a_i}$;
3: Output accept if $F = w^A v^B u^C h^D g^E$, and output reject otherwise.

Obviously, the ABE scheme in [1] is vulnerable to the forgery attack on the outsourced encryption, which can be mounted by the malicious ESP. The basic reason about this security flaw is due to the fact that the product of the ciphertext F_j instead of the ciphertext $(C_{j,1}, C_{j,2}, C_{j,3})$ itself has been regarded as the input of *Batch-Verify* algorithm. It is easily checked that the result returned from *Batch-Verify* algorithm is meaningless: if the *Batch-Verify* algorithm accepts then it need not be the case that the intermediate ciphertext $(C_{j,1}, C_{j,2}, C_{j,3})$ is correct. In other words, any ciphertext with the same product can be successfully batch verified by the *Batch-Verify* algorithm. According to [1], the correctness of *Batch-Verify* algorithm is similar to the small exponents test in [2]. We would like to stress that our attack does not invalidate the security proof for [2], but rather demonstrates how the small exponents test in [2] is exploited in a wrong way. In addition, the security proof of the *Verify.enc* algorithm has not been mentioned in [1] at all and we recommend that the straightforward solution in such *ad-hoc* manner should be avoided in the future.

To clarify the motivation of our attack, we compare the computational cost of our attacks with the original scheme in [1]. We denote by Exp the modular exponentiation in $\mathbb{G}$ and N the number of intermediate ciphertexts involved. Observations from Table 1 demonstrate that both our attacks require less computation than the original scheme given by Ma et al. For financial incentives, the malicious ESP may return incorrect intermediate ciphertexts since these intermediate ciphertext require less computational work and cannot be detected by users.

4 Conclusion

We analyzed the security of an ABE system with verifiable outsourced encryption [1] and showed that any malicious ESP can generate a forged intermediate ciphertext without being detected by the sender. Thus, we remark that it is still challenging to build an ABE system with verifiable outsourced encryption.

Acknowledgments

This work was supported in part by the National Science Foundation of China (No. 61370026, 61672135 and 61602096), Science and Technology Project of Guangdong Province (No. 2016A010101002), Sichuan Science-Technology Support Plan Program (No. 2016JZ0020) and Fundamental Research Funds for the Central Universities (No. ZYGX2016J091).

References

[1] H. Ma, R. Zhang, Z. Wan, Y. Lu, and S. Lin, "Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing," *IEEE Trans. Depend. Secure Comput.*, doi: 10.1109/TDSC.2015.2499755.
[2] M. Bellare, J. A. Garay, and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 1998, pp. 236–250.