

Article

# Attack Algorithm for a Keystore-Based Secret Key Generation Method

Seungjae Chae <sup>1</sup>, Young-Sik Kim <sup>2,\*</sup>, Jong-Seon No <sup>1</sup> and Young-Han Kim <sup>3</sup>

<sup>1</sup> The Department of Electrical and Computer Engineering, Institute of New Media and Communications (INMC), Seoul National University, Seoul 08826, Korea; chae950104@ccl.snu.ac.kr (S.C.); jsno@snu.ac.kr (J.-S.N.)

<sup>2</sup> The Department of Information and Communication Engineering, Chosun University, Gwangju 61452, Korea

<sup>3</sup> The Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093, USA; yhk@ucsd.edu

\* Correspondence: iamyskim@chosun.ac.kr; Tel.: +82-62-230-7032

Received: 19 January 2019; Accepted: 20 February 2019; Published: 23 February 2019

**Abstract:** A new attack algorithm is proposed for a secure key generation and management method introduced by Yang and Wu. It was previously claimed that the key generation method of Yang and Wu using a keystore seed was information-theoretically secure and could solve the long-term key storage problem in cloud systems, thanks to the huge number of secure keys that the keystone seed can generate. Their key generation method, however, is considered to be broken if an attacker can recover the keystore seed. The proposed attack algorithm in this paper reconstructs the keystore seed of the Yang–Wu key generation method from a small number of collected keys. For example, when  $t = 5$  and  $l = 2^7$ , it was previously claimed that more than  $2^{53}$  secure keys could be generated, but the proposed attack algorithm can reconstruct the keystone seed based on only 84 collected keys. Hence, the Yang–Wu key generation method is not information-theoretically secure when the attacker can gather multiple keys and a critical amount of information about the keystone seed is leaked.

**Keywords:** information-theoretically secure; key generation; key management; keystore seed; one-key-for-one-file

## 1. Introduction

Data storage and transmission have been frequently used in recent public cloud systems. It is important to use secure keys in the cloud system, because users using a password can be vulnerable to dictionary attacks [1]. It is well known that secure keys reveal less user information than the password method. Thus, secure keys have been used in various fields such as file encryption, access to virtual private networks, and user authentication [2]. However, conventional key generation methods have many problems in terms of long-term file management, where each file should be independently encrypted with random secure keys since it has the characteristics of long-term file storage and frequent user access. Otherwise, cloud systems are not secure for ciphertext-only attack or chosen-plaintext attack [3]. To make one-key-for-one-file secure encryption for long-term data protection, a new secure key generation method using the keystore seed was proposed in [4] claiming that their method could make many information-theoretically  $\epsilon$ -secure keys. In this paper, we propose a new method to break their key generation by reconstructing the keystore seed using a small number of collected keys.

This paper is organized as follows. In Section 2, the secure key generation and management methods are reviewed. In Section 3, we propose an attack algorithm of the information theoretically  $\epsilon$ -secure key generation method in [4] and show the successful attack probability. In Section 4, we analyze the modified Yang–Wu’s scheme with the hashed key [5], where information is not theoretically  $\epsilon$ -secure, but has only computational security. Finally, Section 5 concludes this paper.

## 2. Key Generation and Management Based on Keystore Seed

In this section, we briefly explain the secure key generation and management methods in [4].

### 2.1. Key Generation

There is a keystore seed  $K = K(0)K(1) \cdots K(L-1)$ , which is a randomly generated  $L$ -bit binary sequence, where  $K(i)$  is the  $i$ -th bit of the keystore seed for  $0 \leq i \leq L-1$ . Let  $a_j$  be a sub-sequence of length  $l$  of the keystore seed and let  $m_j$  be a keystore seed index of the first element of  $a_j$ , where  $0 \leq m_1 < m_2 < \cdots < m_t \leq L-1$ . Then,  $a_j$  is represented as  $a_j = K(m_j)K(m_j+1) \cdots K(m_j+l-1)$ . The key  $k_i$  of length  $l$  is generated as  $k_i = a_1 \oplus a_2 \oplus \cdots \oplus a_t$ , where  $\oplus$  denotes the bit-wise exclusive OR. The set of all possible keys generated from the keystore seed  $K$  is denoted as  $\Psi = \{k_i | 1 \leq i \leq \Lambda\}$ , where  $\Lambda$  is  $\binom{L}{l}$ . This key generation method is expressed as the  $(L, l, t)$ -key generation scheme, where  $l$  is the length of each key and  $t$  is the number of subkeys of keystore seed for the generation of each key.

### 2.2. Key Management

After key generation, the generated keys can be used in the following way:

1. A file is encrypted using a key  $k_i$  randomly selected from set  $\Psi$ .
2. Attach the key index information  $i = (m_1, m_2, \dots, m_t)$  into the encrypted file and send it.
3. To decrypt an encrypted file, the encryption key  $k_i$  is regenerated from the secure stored keystore seed and the received file using  $k_i$  is decrypted using the attached key index information  $i$ .

The keystore seed should be protected in a secure memory that cannot be accessed by outside users. Even though the key index information is available, any information on the keystore seed should not be disclosed.

### 2.3. Information-Theoretically $\epsilon$ -Secure Keystore

The information-theoretically  $\epsilon$ -secure for arbitrarily small  $\epsilon$  is defined according to the following specifications.

**Definition 1** ([4]). A keystore  $\Psi = \{k_i | 1 \leq i \leq \Lambda\}$  of keys of length  $l$  generated from a keystore seed  $K$  is said to be information-theoretically  $\epsilon$ -secure for  $0 \leq \epsilon < 1$ , if the properties in the following theorems hold.

**Theorem 1** ([4]). For  $1 \leq i \leq \Lambda$  and arbitrarily small  $\epsilon > 0$ , all keys  $k_i$  are randomly and uniformly distributed over  $\{0, 1\}^l$  as

$$\Pr\{k_i = k_j\} \leq (1 - \epsilon) \times 2^{-l} + \epsilon.$$

**Theorem 2** ([4]). For all pairs of independent indices  $i, j, 1 \leq i, j \leq \Lambda$ ,

$$H(k_j | i, j, k_i) \geq H(k_j | j) \times (1 - \epsilon) = l(1 - \epsilon).$$

Yang and Wu [4] stated that Theorem 2 can be extended to the following argument.

**Argument 1** ( $n$ -th order of Theorem 2). For all independent  $i, j_1, \dots, j_n$ , where  $1 \leq i, j_1, \dots, j_n \leq \Lambda$ , we have

$$\begin{aligned} H(k_i | j_1, \dots, j_n, i, k_{j_1}, \dots, k_{j_n}) &\geq \\ H(k_i | i) \times (1 - \epsilon) &= l(1 - \epsilon). \end{aligned} \quad (1)$$

In this paper, we will demonstrate that this argument is only true for a very small  $n$ .

### 3. Linear Attack on Key Generation and Management

#### 3.1. Linear Attack Algorithm

In this section, we propose an attack algorithm to reconstruct a keystore seed from a number of collected keys. For example, assume that we have some keys with  $t = 5$  as presented in [4]. Each key has 5 indices and consists of 5 binary exclusive OR subkeys of length  $l$  starting at given indices. Each key can make  $l \times L$  submatrix  $M_i$  shown on the left side of Figure 1. Each  $M_i$  consists of  $l$  indicator vectors to generate key  $k_i$ . For example, we have one key with index  $i = (1, 3, 4, 6, 7)$ . Then, the indicator vector  $e_1^0$  is 0101101100...00 (All 0 except indices 1,3,4,6,7). Next, the indicator vector  $e_1^1$  is a circular shift to the right of  $e_1^0$ . Rows of  $M_i$  consist of  $e_i^0, \dots, e_i^{l-1}$  and  $\text{rank}(M_i) = l$  because it has  $l$  independent indicator vectors. If the  $tl \ll L$  condition is not satisfied, there are dependent indicator vectors due to overlap by cyclic shift. The indicator matrix  $M$  is made by stacking up  $M_i$ 's. Consequently, we stack up submatrices until  $M$  satisfies  $\text{rank}(M) = L$ . Finally, we find keystore seed using the system of linear equations as Figure 1 because  $M$  becomes full rank and it is invertible. The attack algorithm is summarized in Algorithm 1. If the indicator matrix  $M$  has rank  $L$  by stacking up several indicator submatrices, *Argument 1* is not correct for a sufficiently large  $n$  to make  $M$  full rank. Thus, their key generation method is not secure.

---

**Algorithm 1** Successful attack probability with  $R$  keys

---

**Input:** Variables  $L, l, R, t$

**Output:** True if the indicator matrix rank is larger than or equal to  $L$

```

for  $i$  from 1 to  $R$  do
   $key\_index\_set \leftarrow$  Randomly select  $t$  integers in range of  $(0, L - 1)$ 
   $e_i^0 \leftarrow$  indicator vector of  $key\_index\_set$  of length  $l$ 
  for  $j$  from 1 to  $l - 1$  do
     $e_i^j \leftarrow$  circular cyclic shift right once of  $e_i^{j-1}$ 
  end for
   $M_i \leftarrow stack\{e_i^0, \dots, e_i^{l-1}\}$ 
   $M = stack\{M_1, \dots, M_i\}$ 
  if  $\text{rank}(M) \geq L$  then
    return True
  end if
end for

```

---

Let  $Z$  be a random variable defined as

$$Z = \begin{cases} 1 & \text{if } \text{rank}(M) = L \\ 0 & \text{if } \text{rank}(M) \neq L. \end{cases}$$

With this random variable, the left-hand side of (1) can be rewritten as

$$\begin{aligned} H(k_i | j_1, \dots, j_n, i, k_{j_1}, \dots, k_{j_n}) = \\ H(k_i | j_1, \dots, j_n, i, k_{j_1}, \dots, k_{j_n}, Z = 1)P(Z = 1) \\ + H(k_i | j_1, \dots, j_n, i, k_{j_1}, \dots, k_{j_n}, Z = 0)P(Z = 0), \end{aligned} \quad (2)$$

where  $P(Z = 1)$  means that keystore seed is reconstructed and key's entropy goes to 0 because  $k_i$  is automatically determined with key index  $i$ . Therefore, (2) only contains the  $P(Z = 0)$  case. Since  $H(k_i | j_1, \dots, j_n, i, k_{j_1}, \dots, k_{j_n}) \leq l$ , we have

$$H(k_i | j_1, \dots, j_n, i, k_{j_1}, \dots, k_{j_n}, Z = 0)P(Z = 0) \leq lP(Z = 0).$$

According to numerical analysis,  $P(Z = 0)$  becomes almost 0 when the number of collected keys increases, which means that the lower bound of entropy in the  $n$ -th order expansion in *Argument 1* is not correct for a large  $n$ . Although *Argument 1* is correct for very small  $n$ , it is not useful in that they could not generate many secure keys because the purpose of their proposed method is to deal with one-key-for-one-file in cloud systems. In other words, when the entropy of the generated keys becomes 0, the keystore seed cannot be used to generate secure keys anymore. Thus, attackers can reconstruct the keystore seed with high probability, which means that their key generation method is no longer information-theoretically  $\epsilon$ -secure. In the next subsection, we will show the number of collected keys to make  $\text{rank}(M) = L$  by numerical analysis.

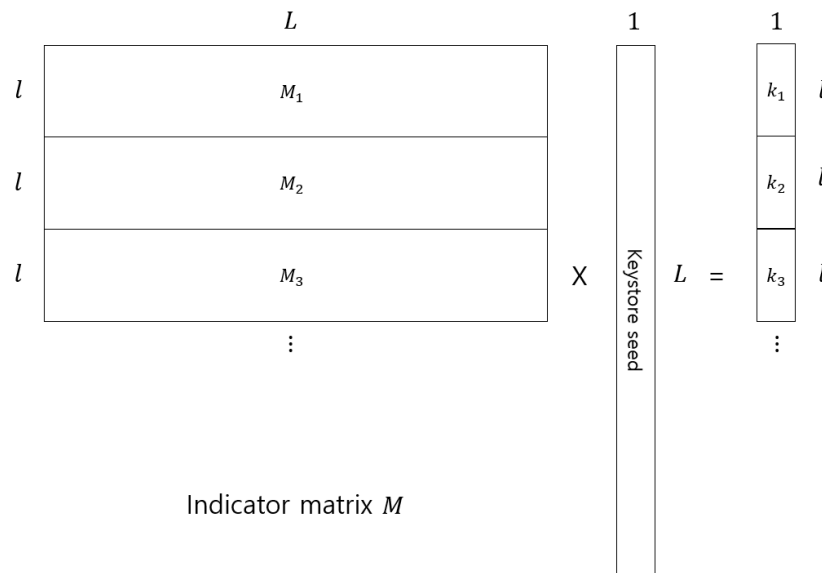
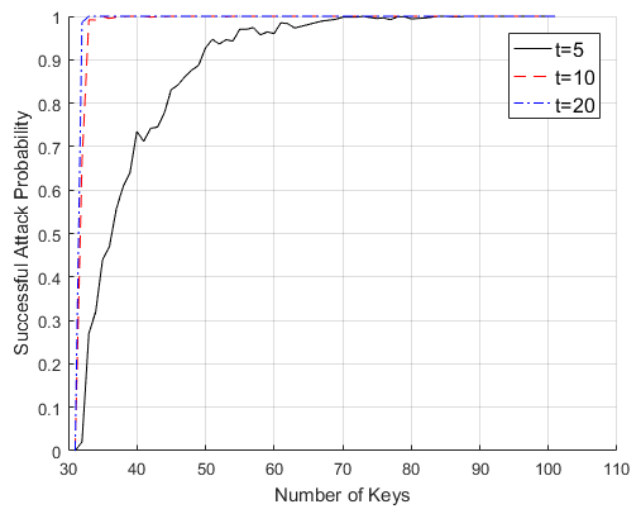


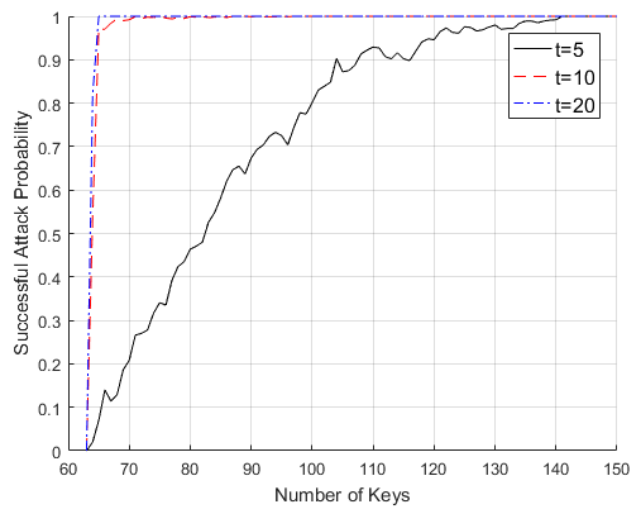
Figure 1. Matrix operation to find keystore seed.

### 3.2. Successful Linear Attack Probability

The successful attack probability with  $R$  keys is given as a probability that an indicator matrix  $M$  has rank larger than or equal to  $L$  by using  $R$  keys as in Algorithm 1. Clearly, at least  $L/l$  keys are required to make  $M$  with full rank. Figure 2 shows that the successful attack probability of the key generation algorithm in [4] is numerically derived for  $L = 2^{12}, 2^{14}, l = 2^7, 2^8$ , respectively, when  $t = 5, 10, 20$ . Table 1 lists the successful attack probability in Figure 2 for several numbers of  $R$ .



(a)



(b)

**Figure 2.** Successful attack probability of the proposed attack algorithm when: (a)  $L = 2^{12}, l = 2^7$ , (b)  $L = 2^{14}, l = 2^8$ .

**Table 1.** Successful attack probability of the proposed attack algorithm.

$L = 2^{12}, l = 2^7$			
Number of keys	$t = 5$	$t = 10$	$t = 20$
$R = 32$	0.02	0.664	0.986
$R = 40$	0.735	1	1
$R = 84$	1	1	1
$L = 2^{14}, l = 2^8$			
Number of keys	$t = 5$	$t = 10$	$t = 20$
$R = 64$	0.02	0.540	0.820
$R = 70$	0.208	0.992	1
$R = 100$	0.801	1	1
$R = 141$	1	1	1

#### 4. Information Theoretic Weakness of Modified Yang-Wu's Schemes with Hashed Keys

The forward secrecy is a property such that if a secret key is compromised, past keys are not compromised. According to the key generation method in [4], several keys are generated from one keystore seed through a linear combination. If the number of generated keys is large enough, the newly generated key will have only a very small entropy from previously generated keys. This idea can be checked via the following observation.

For binary independent random variables  $X$  and  $Y$ , suppose that  $H(X) = H(Y) = 1$  and  $H(X, Y) = 2$ . Then, we have

$$\begin{aligned} H(X, Y|X \oplus Y) &= H(X, Y, X \oplus Y) - H(X \oplus Y) \\ &= H(X, Y) - H(X \oplus Y) = 1. \end{aligned}$$

This can easily be extended and applied to Yang and Wu's algorithm intended to provide independent and uncorrelated secret keys for the one-key-for-one-file long-term secure system. Assume that we have one key generated from  $tl$  bits of keystore seed as in Figure 3. If we know the subkeys  $K(m_j)K(m_j + 1) \cdots K(m_j + l - 1)$  for  $j = 1, \dots, t - 1$ , we can derive the subkey  $K(m_t)K(m_t + 1) \cdots K(m_t + l - 1)$  since we know the key  $k(0)k(1) \cdots k(l - 1)$ . As  $t$  increases, the number of subkeys generating a key becomes large. This becomes a weak point when giving the indicator matrix  $M$  a full rank in Section 3. As the simulation results show that the successful attack probability of the proposed attack algorithm for  $t = 10, 20$  increases abruptly compared to  $t = 5$  when the number of collected keys becomes large. In addition, the successful attack probability becomes very large as  $t$  increases. Therefore, a large value of  $t$  for the key generation scheme should be avoided.

In real applications, it is very important to provide a way of strong protection for the keystore seed. However, in a cloud environment, there is a possibility that some information can be disclosed during the processing such as key generation, file encryption, or decryption, due to undiscovered weakness of systems or side channel attacks as in [6]. In this paper, we show that it is possible to reconstruct the entire keystore seed even if a very small number of generated keys (i.e., 84 keys) are leaked compared to the total size of the possible keys (i.e.,  $2^{53}$  keys).

In order to reduce the risk of keystore seed reconstruction, the encryption using a hashed key  $h(k)$  was proposed in [5], where  $k$  is a generated key from the keystore seed and  $h(\cdot)$  is a one-way hash function. It is true that encryption with a hashed key could avoid the proposed linear attack of keystore seed reconstruction. However, avoiding the linear attack does not guarantee information-theoretically  $\epsilon$ -secure since hashed keys are the same number of bits as original keys. If the original keystore is not information-theoretically  $\epsilon$ -secure, hashed keys are not also information-theoretically  $\epsilon$ -secure since hashing is one to one mapping. Hashing only increases computational complexity, but it does not guarantee key entropy.

The hashed key can be a countermeasure for the proposed linear attack. Moreover, by introducing a hash chain for key generation, it is possible to increase both the computational complexity of the linear analysis and the number of possible keys. Let us set each subkey as  $a_j = K(m_j)K(m_j + 1) \cdots K(m_j + l - 1)$  for  $j = 1, \dots, 5$ . Then, the key  $k_j$  is generated as  $k_j = h(h(h(h(h(a_1) \oplus a_2) \oplus a_3) \oplus a_4) \oplus a_5))$ , where  $a_i$  is a subkey and  $h(\cdot)$  is a one-way hash function from  $\{0, 1\}^*$  to  $\{0, 1\}^l$ . Note that if the order of applying  $a_i$  is changed, the generated key is completely different when a cryptographic hash function such as SHA-2 or SHA-3 is used. Even though this type of countermeasure cannot guarantee information-theoretically  $\epsilon$ -secure keys, but it can be a cryptographically secure way.

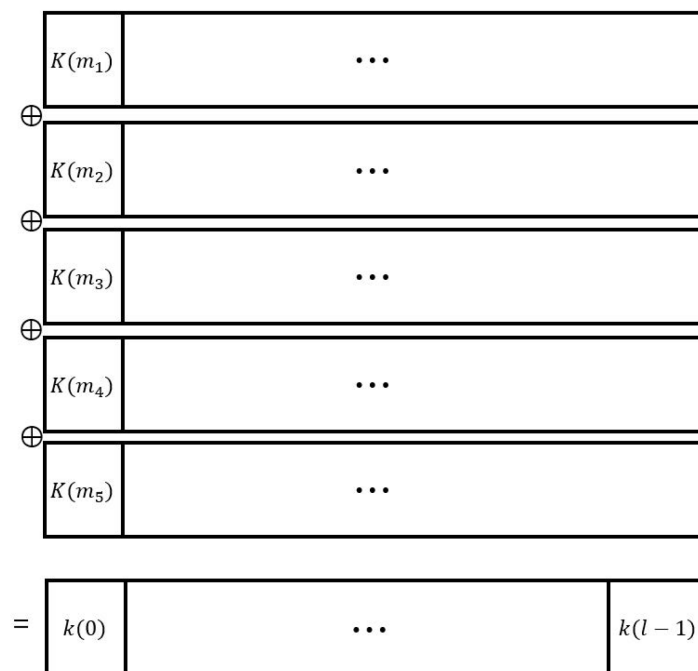


Figure 3. Key generation by subkeys.

## 5. Conclusions

As the demand for long-term data over the public clouds increases, a large number of secure keys are needed. To deal with this problem, Yang and Wu proposed a new key generation method using the keystore seed [4]. In this paper, we proposed an attack algorithm for their key generation method, where a small number of collected keys can be used to reconstruct the keystore seed with high probability. Although the encryption using a hashed key could avoid the proposed reconstruction attack, it still does not guarantee the information-theoretically  $\epsilon$ -secure in certain situations where some information is leaked. Therefore, a new secure key generation method with keystore seed can be studied in future research.

**Author Contributions:** S.C. firstly found the main issue of the previous scheme. Y.-H.K. proposed a methodology to analyze this issue. All authors carried out the formal analysis of the proposed attack. S.C. wrote a program in C and Y.-S.K. and J.-S.N. investigated the numerical data. All authors have read and approved the final manuscript.

**Funding:** This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (R-20160229-002941, Research on Lightweight Post-Quantum Crypto-systems for IoT and Cloud Computing).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Morris, R.; Thompson, K. Password security: A case history. *Commun. ACM* **1979**, *22*, 594–597. [[CrossRef](#)]
2. Monrose, F.; Reiter, M.K.; Li, Q.; Wetzels, S. Cryptographic key generation from voice. In Proceedings of the 2001 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 14–16 May 2001; pp. 202–213.
3. Menezes, A.J.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
4. Yang, E.H.; Wu, X.W. Information-theoretically secure key generation and management. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017; pp. 1529–1533.

5. Wu, X.W.; Yang, E.H.; Wang, J.H. Lightweight security protocols for the Internet of Things. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017.
6. Bazm, M.-M.; Lacoste, M.; Sudholt, M.; Menaud, J.-M. Side Channels in the Cloud: Isolation Challenges, Attacks, and Countermeasures. 2017. Available online: <https://hal.inria.fr/hal-01591808/> (accessed on 17 February 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).