Double Encryption for Data Authenticity and Integrity in Privacy-preserving Confidential Forensic Investigation

Shuhui Hou*
University of Science and Technology Beijing
Beijing, China
shuhui@ustb.edu.cn

Ryoichi Sasaki
Tokyo Denki University
Tokyo, Japan
sasaki@im.dendai.ac.jp

Tetsutaro Uehara
Ritsumeikan University
Kyoto, Japan
uehara@cs.ritsumei.ac.jp

Siuming Yiu
The University of Hong Kong
Hong Kong, China
smyiu@cs.hku.hk

Abstract

It is getting popular that users will put their data in cloud computing services or data centers. It applies to criminals too. In such computing platforms, data will be stored in large servers. In other words, evidence for crime cases may exist in a large storage media or even distributed in various storage device(s) that may be in different sites. The traditional approach of cloning a copy of data in forensic investigation will not work. Besides, those users irrelevant to the crime are not willing to disclose their private data for investigation. To solve these problems, Hou et al. provided the first solutions to let the server administrator (without knowing the investigation subject) to retrieve only the data that is relevant to the cases based on the technique of searching encrypted keywords over encrypted data. In this case, the privacy data of irrelevant users can be protected from disclosing. However, in their solutions, it is no way to confirm the authenticity and integrity of the collected data. This is critical when presenting the evidence to court. In this paper, we try to tackle this problem and provide a solution to verify the authenticity and integrity of the evidence in addition to the security requirements for privacy-preserving confidential forensic investigation. Our solution is based on a "double encryption" scheme. We provide a security analysis of the scheme and we also implemented the proposed scheme based on RSA cryptosystem. Experimental results show that the performance of the scheme is reasonable.

Keywords: confidential forensic investigation, authenticity and integrity, commutative encryption

1 Introduction

With the rapid development of various computing platforms such as cloud computing, high-speed computing centers, many users including criminals will put their data in these platforms. In other words, if a crime case occurs, forensic investigators have to apply a warrant and try to retrieve evidence from the servers of these platforms. However, there are two issues that make this investigation extremely difficult. First, the traditional method of disk cloning will not work due to the massive volume of the data and these storage devices may be distributed in different locations (even outside the region that the crime case occurs). Second and more importantly, the storage devices store not only the data of the suspect, but also an enormous amount of data involving possibly sensitive information that is totally irrelevant to the crimes [1]. These innocent users have the right not to release or disclose their data to the investigator

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 4, number: 2, pp. 104-113

*Corresponding author: School of Mathematics and Physics, University of Science and Technology Beijing, 30 Xueyuan Road, Haidian District, Beijing 100083, China, Tel: +86-15910667849

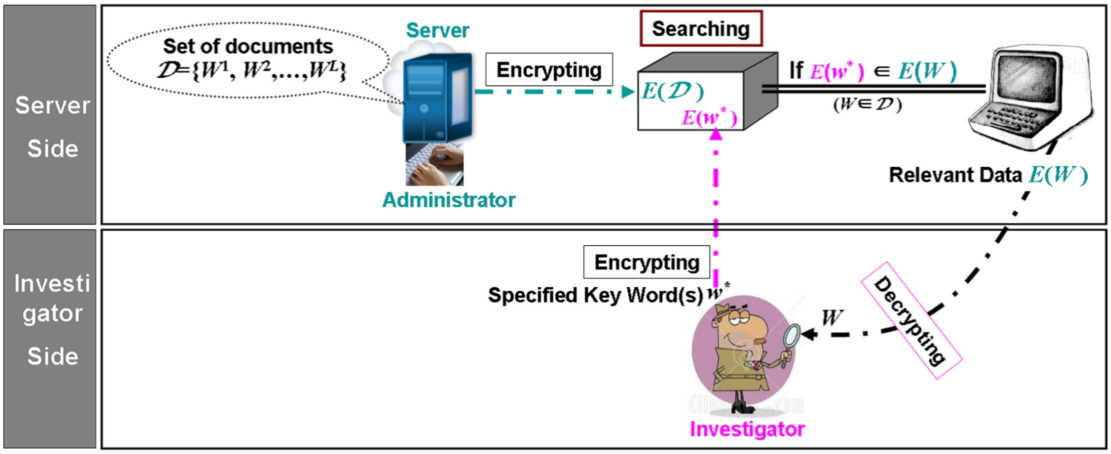

Figure 1: Searching on Encrypted Data with Encrypted Keyword [2]

as the data may be confidential (e.g. related to financial investment data). To improve the investigation efficiency and protect the privacy of irrelevant users, a naive approach is to let the server administrator search, retrieve and hand only the relevant data to the investigator, where the administrator is supposed to be responsible for managing the data in a secure manner. However, this is not feasbile as there are crime cases that must be kept confidential until enough evidence is found. In other words, the investigator wants to keep the subject to be identified confidential so that even the administrator cannot know what he is looking for. So, the problem is how to protect both confidentiality of investigation and privacy of irrelevant users at the same time. We refer to this problem as "server-aided privacy-preserving confidential forensic investigation".

Hou et al. [2, 3] is the first group that provided solutions to solve this problem. We will follow the same assumption in their solutions that the server administrator is willing to cooperate with the investigator to search the relevant data. Their solutions are based on encrypted keyword searching over encrypted data (shown in Figure 1). The high level workflow is as follows. (1) the investigator encrypts the keyword(s) that is (are) relevant to the crime case with his public key and sends the encrypted keyword(s) to the administrator; (2) with the investigator's public key, the administrator encrypts all the data files stored on the server. Then, he uses the encrypted keyword(s) to search over encrypted data files, retrieves and sends only the relevant data (i.e., those encrypted files whose corresponding plaintext files contain the specified keyword(s)) to the investigator; (3) the investigator decrypts the relevant data with his private key and performs investigation only on such relevant data for collecting the criminal evidence. The irrelevant data (those files without containing the keyword(s)) will never be sent to the investigator, so can be protected from exposing to the investigator. By using encrypted keyword(s) to search over encrypted data, the administrator has no idea of what the investigator is looking for.

In the above solutions, the administrator can protect the irrelevant data against disclosing, and at the same time he is prevented from learning what the investigator is looking for. In order for the administrator to check what data (whether they are really relevant) the investigator obtains for investigation, the solutions in [2, 3] assume that the administrator can require the investigator to show what data is collected based on what keyword(s) when the relevant data is presented as evidence in court. However, there is still a major problem in this. Even it is possible to let the administrator look at the data presented to court, no measures can guarantee that the presented data is the one that comes from the server without alteration. In other words, the authenticity and integrity of the evidence collected in the work [2, 3] are not considered. The authenticity and integrity are two fundamental requirements for admissibility of evidence in court and they are crucial to win a case [4], but were not addressed in existing solutions. To date, several methods from the computer science and information security have been adapted to prove the integrity of digital evidence, for example, checksum, one-way hash algorithm, digital signature, and so on [5]. In this paper, we will solve the above-mentioned problem and propose "double encryption" scheme to prove the authenticity and integrity of the evidence collected in the work [2, 3].

In this paper, we propose a "double encryption with designated verifier" scheme to prove the authenticity and integrity of the evidence. When the above-mentioned relevant data is presented as evidence during a trial, our scheme enables the administrator (or the third party the administrator trusts) to verify whether the presented evidence actually comes from the server without alternation. We show that our scheme is secure and also demonstrate its performance by implementing the scheme and evaluating it by experiments.

The rest of the paper is organized as follows. In Section 2, we introduce some preliminary knowledge such as commutative encryption. We define the requirements of the problem and give the details of our "double encryption with designated verifier" scheme using commutative encryption in Section 3. Section 4 evaluates the security and performance of our scheme. Finally, Section 5 concludes the paper.

2 Preliminaries: Commutative Encryption

We first talk about commutative encryption.

Definition 1 Let $\mathscr{M}$ denote a message space, $\mathscr{K}$ denote a key space and $\mathscr{C}$ denote a cipher message space, respectively. A commutative encryption function is a family of bijections $\mathscr{E}$: $\mathscr{M}\times\mathscr{K}\rightarrow\mathscr{C}$ such that for a given $m\in\mathscr{M}$ we have $\mathscr{E}_{k_1}(\mathscr{E}_{k_2}(m))= \mathscr{E}_{k_2}(\mathscr{E}_{k_1}(m))$, for any $k_1,k_2\in\mathscr{K}$.

It follows that if a message is encrypted by two different keys k_1 and k_2, it can be recovered by decrypting the cipher message with k_1 followed by decrypting with k_2. The message can also be recovered by decrypting with k_2 followed by decrypting with k_1.

The RSA cryptosystem (introduced in 1978, [6]) is commutative for keys with a common modulus n. One description of the system is given below.

RSA Cryptosystem. Let $n=pq$ where p and q are a pair of large, random primes. Select e and d such that $ed=1 \pmod{\phi(n)}$ where $\phi(n)=(p-1)(q-1)$. n and e are public while p,q and d are private.
The encryption operation is:

$$c = ENCRYPT(m) = m^e \mod n$$

The decryption operation is:

$$m = DECRYPT(c) = c^d \mod n$$

Where m is the plaintext message and c is the resulting ciphertext.

Using $\mathscr{E}_k(\cdot)$ to denote the encryption operation with key k, it is obvious that

$$\begin{aligned}
\mathscr{E}_{e_1}(\mathscr{E}_{e_2}(m)) &= (m^{e_2})^{e_1} \bmod n \\
&= m^{e_2 e_1} \bmod n \\
&= m^{e_1 e_2} \bmod n \\
&= (m^{e_1})^{e_2} \bmod n \\
&= \mathscr{E}_{e_2}(\mathscr{E}_{e_1}(m)) \bmod n
\end{aligned}$$

i.e., the RSA cryptosystem is commutative for keys with a common modulus n.

3 Proposed Scheme: Double Encryption with Designated Verifier

3.1 Requirements of "Server-aided Privacy-preserving Confidential Forensic Investigation"

We first list the the requirements for solving the server-aided privacy-preserving confidential forensic investigation problem below.

From the **investigator's** viewpoint, he hopes to fulfill the followings:

- Collect evidence only from relevant data for saving time and effort, so as to improve the investigation efficiency;
- Let server administrator search and retrieve relevant data but without letting him know what he is searching and retrieving;
- Verify the authenticity and integrity of the relevant data so that it can be admitted in court when it is presented as evidence.

From the **administrator's** viewpoint, he hopes to fulfill the followings:

- Protect irrelevant data against exposing while cooperating with investigator in collecting evidence, i.e., ensuring that no privacy of irrelevant users leaks during investigation;
- Be able to verify the authenticity and integrity of the relevant data when it is presented as evidence in court.

3.2 Our Proposed Scheme

For ease understanding, we consider a single keyword in the following. But the solution can be easily extended to multiple keywords. We denote the single keyword specified by the investigator as w^*, which is l-bit long; The data stored on the server is assumed to be a set of documents, denoted as $\{W^1, W^2, \ldots, W^L\}$. A document $W \in \{W^1, W^2, \ldots, W^L\}$ consists of a sequence of words, denoted as $W=\{w_1, w_2, \ldots, w_v\}$ where every word w_i is l-bit long. We also assume that both w^* and W come from the same domain. It should be pointed out that a document does not always consist of equal-length words, but we can transform the variable-length words into fixed-length words through hashing. The encryption of w^* and W is denoted as $\mathscr{E}(w^*)$ and $\mathscr{E}(W)=\{\mathscr{E}(w_1), \mathscr{E}(w_2), \ldots, \mathscr{E}(w_v)\}$, where $\mathscr{E}(\cdot)$ is the encryption function.

Assume that there is a secure channel between server administrator and investigator. Based on commutative encryption, the "double encryption with designated verifier" scheme works as follows.

The first encryption for confidentiality and privacy: For the confidentiality of investigation, the investigator encrypts his specified keyword w^* with his public key p_I and sends the administrator the encrypted keyword $\mathscr{E}_{p_I}(w^*)$ as well as his public key p_I; on server side, the administrator encrypts all the documents $\{W^1, W^2, \ldots, W^L\}$ with the public key p_I, where the resulting documents are denoted as $\{\mathscr{E}_{p_I}(W^1), \mathscr{E}_{p_I}(W^2), \ldots, \mathscr{E}_{p_I}(W^L)\}$. Note that the keyword and also the documents are encrypted. The administrator is not able to know what keyword the investigator wants and the investigator also cannot know anything not related to the keyword specified. This achieves the requirements of confidentiality and privacy. Besides, we require that the administrator cannot build the correspondence between original documents and encrypted documents. This can be realized by setting up a black-box on server side. In the black-box, the documents are shuffled before encryption, which can prevent the administrator from seeing plaintext and ciphertext pairs.

Double encryption for authenticity and integrity: On server side, the administrator performs the following. He uses $\mathscr{E}_{p_I}(w^*)$ to search over all the encrypted documents $\{\mathscr{E}_{p_I}(W^1), \mathscr{E}_{p_I}(W^2), \ldots, \mathscr{E}_{p_I}(W^L)\}$, and retrieves $\mathscr{E}_{p_I}(W)$ such that the plaintext document W contains the keyword w^* (i.e., $w^*{\in}W$). There are several approaches to check if the plaintext document W contains the keyword w^* based on the relation between the ciphertext $\mathscr{E}(W)$ and $\mathscr{E}(w^*)$. As $\mathscr{E}(\cdot)$ is a deterministic encryption (e.g., RSA cryptosystem), we can get $w^*{\in}W$ if $\mathscr{E}(w^*){\in}\mathscr{E}(W)$, i.e., there exist one word $w_i \in W$ such that $\mathscr{E}(w_i){=}\mathscr{E}(w^*)$; as $\mathscr{E}(\cdot)$ is a probabilistic encryption, $w^*{\in}W$ can be shown by zero-knowledge proof (please refer to the work [2] where Paillier cryptosystem [7, 8] is used). Note that using the probabilistic encryption, the same plaintext will not give the same ciphertext on different encryption. Thus, even if the attacker sees some plaintext and ciphertext pairs before, next time, when the investigator searches for the same plaintext, the ciphertext will not be the same.

To verify the authenticity and integrity, the administrator encrypts W again by computing $\mathscr{E}_A(\mathscr{E}_{p_I}(W))$ if $w^*{\in}W$ and sends the investigator $\mathscr{E}_{p_I}(W)$ as well as $\mathscr{E}_A(\mathscr{E}_{p_I}(W))$, where $\mathscr{E}_A(\cdot)$ is commutative encryption with $\mathscr{E}_{p_I}(\cdot)$ and the subscript 'A' means that it is the administrator's encryption function. As $\mathscr{E}_A(\cdot)$ is public key encryption, $\mathscr{E}_A(\mathscr{E}_{p_I}(W))$ means encrypting $\mathscr{E}_{p_I}(W)$ with the public key of the administrator. As $\mathscr{E}_A(\cdot)$ is secret key encryption, the secret key is only known to the administrator. In the following, we consider that $\mathscr{E}_A(\cdot)$ is public key encryption. Here, the other documents without containing the keyword w^* will never be sent to the investigator, so their privacy can be protected completely.

This second encryption has the following properties that make it suitable for our application.

- **Selective encryption**: the administrator encrypts only the relevant data W ($w^*{\in}W$) instead of all the data stored on the server for less computational cost.

- **Blind encryption**: the administrator wants to verify the authenticity and integrity of the original relevant data W ($w^*{\in}W$) rather than its encrypted form $\mathscr{E}_{p_I}(W)$, so conceptually he needs to encrypt W blindly, that is, encrypt W without knowing what W is. Here, the administrator encrypt W blindly by computing $\mathscr{E}_A(\mathscr{E}_{p_I}(W))$, i.e., computing encryption of W twice.

- **Designated verifier**: the administrator wants to check if the relevant data is really *relevant* to the crimes and ensure that the investigator does not obtain other irrelevant data from the server. The administrator needs to control the encryption verification. On the other hand, the investigator also needs the administrator's cooperation to prove that the relevant data does come from the server without alteration when it is presented as evidence. In a word, a designated verifier is required here. In our scheme, only the administrator can confirm the integrity and can authenticate the data. The administrator can also delegate the verification key to the third party he trusts and let the third party verify the data.

Decryption: The investigator decrypts $\mathscr{E}_{p_I}(W)$ ($w^*{\in}W$) with his private key and performs investigation on W for capturing evidence. He also decrypts $\mathscr{E}_A(\mathscr{E}_{p_I}(W))$(which is $\mathscr{E}_{p_I}(\mathscr{E}_A(W))$ as $\mathscr{E}_{p_I}(\cdot)$ and $\mathscr{E}_A(\cdot)$ are commutative) for obtaining the encrypted W, i.e., $\mathscr{E}_A(W)$. He keeps the $\mathscr{E}_A(W)$ for the later verification.

Authenticity and integrity verification: When W is presented as evidence in court, the administrator (or the third party the administrator trusts) verifies the data by testing if $\mathscr{D}_A(\mathscr{E}_A(W)){=}W$ is true, where $\mathscr{D}_A(\cdot)$ is the inverse of encryption process $\mathscr{E}_A(\cdot)$. In other words, the administrator (or the third party the administrator trusts) verifies if the evidence is W that comes from the server without alteration. Note that we use RSA in an unconventional way, for details, please refer to Section 4.2, thus the simple attack of replacing W by W' using the conventional RSA will not work.

4 Security Analysis and Implementation

4.1 Security Analysis

As described above, our “double encryption with designated verifier” scheme can prove the authenticity and integrity of the evidence collected in the work [2, 3].

1. In our scheme, it is not possible for someone else to replace W with his own since only the administrator (or the third party the administrator trusts) knows the verification key. That is, only the administrator (or the third party the administrator trusts) can verify the data correctly. This fact can also be used to prove that the presented evidence is the relevant data from the server(i.e., proof to data authenticity);

2. Benefiting from commutative encryption, the administrator can blindly encrypt the original relevant data. By verifying the encryption, the administrator (or the third party the administrator trusts) can check if alteration occurs to the presented evidence or not (i.e., proof to data integrity).

   To prove the data authenticity and integrity is crucial for that the relevant data can be admitted as evidence in court.

3. In comparison with the existing work [2, 3], our scheme can ensure the server data against unauthorized disclosing by double encryption. It is provably secure in the sense that the administrator cannot learn anything about the investigation subject and the investigator cannot learn more than the searching results.

   Similar to the work in [2, 3], there is an underlying assumption in our scheme: the investigator trusts the administrator to return all the searching results if they satisfy the searching criteria of the investigator. This ‘trust’ is similar to our trust in postman, i.e., we believe that the postman will send letters without secretly opening the letter. Since the processes of encryption, searching and retrieving are conducted on server side, it is possible for the administrator to figure out the investigation subject by mapping the searching results with the plaintext data. If the investigator does not trust the administrator at all, the above-mentioned encryption, searching and retrieving processes can be conducted under the supervision of both investigator and administrator. That is, the investigator has to participate in exclusion of irrelevant data. Regarding the case of postman, it is possible for the postman to secretly open the letter during mailing. We have to send letters by ourselves if we do not trust the postman at all. Simply, this ‘trust’ involves a tradeoff between security and convenience. However, whether the investigator trusts the administrator to return all the searching results or not, the data authenticity and integrity needs to prove since searching results are in encrypted form. The administrator has no idea of what data is retrieved during investigation, so he needs to check it later for ensuring that the server data is used in a right way. On the other hand, the investigator also needs to prove the data authenticity and integrity so that the data can be admitted as evidence in court. That is the point why we conduct this research.

4.2 Implementation

We apply the commutative encryption to implement our “double encryption with designated verifier” scheme. We use the RSA cryptosystem in an unconventional way. The investigator chooses primes p and q, computes $n{=}pq$ and chooses an RSA key pair e_I and d_I, where e_I and n are public while p, q and d_I are private. Using the same n, the administrator also chooses an RSA key pair e_A and d_A such that $e_A d_A{=}1 \bmod \phi(n)$ but keeping both e_A and d_A secret. The way RSA is usually used, n and e are public, and it is believed to be hard to find m given $c{=}m^e \bmod n$. Here, we consider it may be hard to find e_A and d_A even knowing the factorization of n [9]. For security, we need more than just that it is hard to find e_A and d_A, but this is not our focus. The security of our scheme relies on what commutative encryption we will use. To improve the security of our system, we can apply the commutative encryption with high security.

We carry out implementation on the Genuine Intel(R) CPU U7300, 1.30 GHz PC with RAM 2.00 GB, MATLAB 7 as integrated environment. We take three Microsoft word documents (consisting of English words which are separated by space) and set parameters of RSA cryptosystem as follows: $p{=}29$, $q{=}37$, $n{=}1073$, $e_I{=}5$, $d_I{=}605$, $e_A{=}11$ and $d_A{=}275$.

Table 1 lists all the execution CPU time in seconds (s), which are average time we randomly set the position of the keyword in a document five times. The items in Table 1 are detailed below.

- “W (words)”: three word documents are used in implementation, which consist of 250, 1000 and 5000 English words respectively;
- “Preprocess W”: since the documents consist of variable-length words, the administrator uses MD5 hash function to transform them into fixed-length words so that each document meets the assumption mentioned in Section 3.2. In practice, we can replace MD5 by SHA-256 in the scheme if we want to increase the security level. This will affect the preprocessing time (it is estimated to be about three times longer);
- “Compute $\mathscr{E}_{e_I}(W)$”: the administrator encrypts each document with the investigator’s public key e_I;
- “Search $\mathscr{E}_{e_I}(W)$”: the administrator uses the encrypted keyword $\mathscr{E}_{e_I}(w^*)$ to search over encrypted documents and retrieve $\mathscr{E}_{e_I}(W)$ if W contains w^* (i.e., $w^*{\in}W$).
  RSA cryptosystem is deterministic encryption, we can get $w^*{\in}W$ if $\mathscr{E}(w^*){\in}\mathscr{E}(W)$, that is, we can get $w^*{\in}W$ if there exist one word $w_i \in W$ such that $\mathscr{E}(w_i){=}\mathscr{E}(w^*)$;
- “Compute $\mathscr{E}_{e_A}(\mathscr{E}_{e_I}(W))$”: for the $\mathscr{E}_{e_I}(W)$ such that $w^*{\in}W$, the administrator encrypts $\mathscr{E}_{e_I}(W)$ with his key e_A for blindly encrypting the document W;
- “Decrypt $\mathscr{E}_{e_I}(W)$”: the investigator decrypts $\mathscr{E}_{e_I}(W)$ ($w^*{\in}W$) with his private key d_I to obtain W for investigation;
- “Decrypt $\mathscr{E}_{e_A}(\mathscr{E}_{e_I}(W))$”: the investigator decrypts $\mathscr{E}_{e_A}(\mathscr{E}_{e_I}(W))$ ($=\mathscr{E}_{e_I}(\mathscr{E}_{e_A}(W))$ as $\mathscr{E}_{e_I}(\cdot)$ and $\mathscr{E}_{e_A}(\cdot)$ are commutative) with his private key d_I to obtain $\mathscr{E}_{e_A}(W)$ ($w^*{\in}W$);
- “Decrypt $\mathscr{E}_{e_A}(W)$”: when the W ($w^*{\in}W$) is presented as evidence in court, the administrator decrypts $\mathscr{E}_{e_A}(W)$ with his verification key d_A and cooperates with the investigator in verifying if $\mathscr{D}_{d_A}(\mathscr{E}_{e_A}(W)) = W$.

For the RSA cryptosystem, let l denote the size of the modulus n in bits, i.e., $l{=}[\log_2 n]$. The private exponent d is of similar size while the public exponent e is usually some small number, and the size of plaintext m is limited by $O(l)$. Encryption and decryption are both modular exponentiations of plaintext and ciphertext modulo n with the respective exponents. Using the typical modular exponentiation algorithms to implement the RSA cryptosystem, the computational complexity of encryption and decryption operations are $O(l^2)$ and $O(l^3)$ respectively. The CPU time on encryption and decryption operations are shown in Table 1. It is obvious that the execution CPU time increases as the size of document is growing. The reason consists in that most of processes are based on encryption and decryption of RSA cryptosystem, which are time consuming on large amount of data. Here, we can reduce the time complexity by applying parallel computation of modular exponentiation.

Table 1: Experimental Results

W (words)	Administrator Side				Investigator Side		Court
	Preprocess W (s)	Compute $\mathscr{E}_{e_I}(W)$ (s)	Search $\mathscr{E}_{e_I}(W)$ (s)	Compute $\mathscr{E}_{e_A}(\mathscr{E}_{e_I}(W))$ (s)	Decrypt $\mathscr{E}_{e_I}(W)$ (s)	Decrypt $\mathscr{E}_{e_A}(\mathscr{E}_{e_I}(W))$ (s)	Decrypt $\mathscr{E}_{e_A}(W)$ (s)
250	1.5756	1.8720	0.2652	2.0280	2.9172	2.7300	2.7300
1000	5.9124	7.3944	0.2964	7.9873	11.4661	11.1385	10.9357
5000	27.8930	36.7850	0.3744	40.1457	58.0480	54.9748	54.7408

5 Discussion and Conclusions

5.1 Discussion

Because the public key encryption is much less efficient than the secret key encryption, we can use a secret key encryption to replace the above RSA cryptosystem for blind encryption, where the same secret key is used to encrypt and verify, and it is only known to the administrator (or the third party the administrator trusts). Using the secret key encryption can improve the efficiency, but it is hard to construct a secret key encryption which is commutative with the public key encryption. So, we still need to consider how to realize our scheme by a more general way.

Our "double encryption with designated verifier" scheme provides a practical solution to the "server-aided privacy-preserving confidential forensic investigation". First, in the area of digital forensics and e-discovery, using keyword searching is currently the most utilized and the most widely recognized culling method while not perfect; Second, encryption and decryption operations may be time-consuming, but it can be solved by applying parallel computation of modular exponentiation. Our work will have a wide application prospect because all processes of "double encryption with designated verifier" on server side can be performed fully automatically.

A preliminary version of this paper was presented at the ICT-EurAsia, March 2013 [10]. The main difference between this paper and the preliminary version consists in: we rephrased the research problem, performed security analysis and implemented the proposed "double encryption" in this paper. Experimental results show that the performance of our scheme is reasonable.

5.2 Conclusions and Future Work

This paper proposed "double encryption with designated verifier" scheme to verify the authenticity and integrity of the evidence collected in [2, 3]. The proposed scheme can assist the administrator in checking whether the investigator cheats on obtaining irrelevant data from the server and whether alteration occurs to the evidence or not. Hence, both confidentiality of investigation and privacy of irrelevant users can be protected in "server-aided confidential forensic investigation". In addition, we implemented the proposed scheme and analyzed its security. For future work, we will put the proposed system into practice by working with some local law enforcement units to furher evaluate its feasibility. Also, we should develop multi-dimensional search (e.g., range search, equality search, etc.) over encrypted data to overcome the restriction of the keyword search.

5.3 Acknowledgments

This work is partially supported by "Heiwa Nakajima Foundation, Japan" and partially sponsored by "the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry".

References

[1] A. Aminnezhad, A. Dehghantanha, and M. T. Abdullah, "A survey on privacy issues in digital forensics," *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, vol. 1, no. 4, pp. 311–323, 2012.

[2] S. Hou, T. Uehara, S. Yiu, L. C. Hui, and K. Chow, "Privacy preserving confidential forensic investigation for shared or remote servers," in *Proc. of the 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'11), Dalian, China*. IEEE, October 2011, pp. 378–383.

[3] ——, "Privacy preserving multiple keyword search for confidential investigation of remote forensics," in *Proc. of the 3rd International Conference on Multimedia Information Networking and Security (MINES'11), Shanghai, China*. IEEE, November 2011, pp. 595–599.

[4] Y. Kim, "Digital forensics formats: Seeking a digital preservation storage container format for web archiving," *The International Journal of Digital Curation*, vol. 7, no. 2, pp. 21–39, 2012.

[5] C. Hosmer, "Proving the integrity of digital evidence with time," *International Journal of Digital Evidence*, vol. 1, no. 1, pp. 1–7, 2002.

[6] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 2, no. 21, pp. 120–126, 1978.

[7] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. of the 17th international conference on Theory and application of cryptographic techniques (EUROCRYPT'99), Prague, Czech Republic, LNCS*, vol. 1592. Springer-Verlag, May 1999, pp. 223–238.

[8] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP Journal on Information Security*, vol. 2007, pp. 1–10, 2007.

[9] A. Yamamura, "Asymmetric secret key ciphers and encrypted data retrieval schemes," *RIMS Kokyuroku*, vol. 1562, pp. 73–78, 2007.

[10] S. Hou, R. Sasaki, T. Uehara, and S. Yiu, "Verifying data authenticity and integrity in server-aided confidential forensic investigation," in *Proc. of Information & Communication Technology-EurAsia Conference 2013 (ICT-EurAsia'13), Yogyakarta, Indonesia, LNCS*, vol. 7804. Springer-Verlag, March 2013, pp. 312–317.

Shuhui Hou received the B.S. and M.S. degree in Mathematics from China, in 1993 and 1996, respectively. She received the Ph.D. in Informatics from Kyoto University Japan in 2009. She has been with University of Science and Technology Beijing, China since 1996. Her research interests are in information security and digital forensics.

Ryoichi Sasaki is a professor of Dept. of Information Systems and Multi Media, School of Science and Technology for Future Life, Tokyo Denki University. He received his B.S. Degree in health science and Ph.D. Degree in system engineering, both from the University of Tokyo in 1971 and 1981, respectively. From April of 1971 to March of 2001, he was engaged in the research and research management on systems safety, network management and information security at Systems Development Laboratory of Hitachi Ltd. From April of 2001, he is a professor of Tokyo Denki University, and engaged in the research and education on information security. Now, he is also an advisor of Information Security in Cabinet Secretariat for Government of Japan, a visiting professor of National Institute of Informatics, and a general chair of Japan Society of Security Management.

Tetsutaro Uehara received the B.E, M.E., and Ph.D degrees from Kyoto University in 1990, 1992 and 1996, respectively.From 1994 to 1995, he was a research fellow of the Japan Society for the Promotion of Science (JSPS). From 1995 to 1996, he was a research associate of the Graduate School of Engineering, Kyoto University. From 1997 to 2003, he was an assistant professor of the Faculty of Systems Engineering, Wakayama University. He was also an assistant professor of the Center for Information Science, Wakayama University from 1997 to 2000. From 2003 to 2005, he was an associate professor at the Center for Information Technology of the Graduate School of Engineering, Kyoto University. From 2006 to 2011, he was an associate professor at Academic Center for Computing and Media Studies, Kyoto University. From 2011 to 2013, he was a Senior Deputy Director at the Standardization Division and ICT Security office of Ministry of Internal Affairs and Communication, Japan. Since 2013, he is a Professor at College of Information Science and Engineering, Ritsumeikan University. His research covers ICT System Security, System Management and Digital forensics.

Siuming Yiu is currently an Associate Professor of the Department of Computer Science at the University of Hong Kong. He graduated with his PhD in Computer Science in the same department in 1997. His areas of research interest include bioinformatics, cryptography, and computer security. He is the deputy executive director of the HKU-BGI BAL (Bioinformatics Algorithms and Core Technology Research Laboratory) and the leader of the Applied Cryptography Research Group of the Center for Information Security and Cryptography.