Verifying Data Authenticity and Integrity in Server-Aided Confidential Forensic Investigation

Shuhui Hou[1], Ryoichi Sasaki[2], Tetsutaro Uehara[3], and Siuming Yiu[4]

[1] Department of Information and Computing Science, University of Science and Technology Beijing, China
[2] Graduate School of Science and Technology for Future Life, Tokyo Denki University, Japan
[3] Research Institute of Information Security, Wakayama, Japan
[4] Department of Computer Science, The University of Hong Kong, Hong Kong

Abstract. With the rapid development of cloud computing services, it is common to have a large server shared by many different users. As the shared server is involved in a criminal case, it is hard to clone a copy of data in forensic investigation due to the huge volume of data. Besides, those users irrelevant to the crime are not willing to disclose their private data for investigation. To solve these problems, Hou et al. presented a solution to let the server administrator (without knowing the investigation subject) cooperate with the investigator in performing forensic investigation. By using encrypted keyword(s) to search over encrypted data, they realized that the investigator can collect the necessary evidence while the private data of irrelevant users can be protected from disclosing. However, the authenticity and integrity of the collected evidence are not considered there. The authenticity and integrity are two fundamental requirements for the evidence admitted in court. So in this paper, we aim to prove the authenticity and integrity of the evidence collected by the existing work. Based on commutative encryption, we construct a blind signature and propose a "encryption-then-blind signature with designated verifier" scheme to tackle the problem.

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

1 Introduction

With the rapid development of cloud computing technology, forensic investigation is becoming more and more difficult as crimes occur. The traditional technique disk cloning may be impossible to conduct for collecting evidence from a data center due to the massive volume of data and the distributed manner of storage device(s). Besides, it is common to have a large server shared by many different users in the cloud computing environment. The shared server stores not only suspicious data relevant to the crimes but also stores an enormous amount of data involving *sensitive* information that is totally irrelevant to the crimes. The users irrelevant to the crimes may not want to release their information

K. Mustofa et al. (Eds.): ICT-EurAsia 2013, LNCS 7804, pp. 312–317, 2013.

for investigation especially as it involves confidential or privacy information. To improve the investigation efficiency and protect the privacy of irrelevant users, one strategy 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. Due to some special crimes, the investigator may not want the administrator to know what he is looking for. In short, it is indispensible to consider how to protect both confidentiality of investigation and privacy of irrelevant users in such forensic investigation. For simplicity of description, we refer to this problem as "server-aided confidential forensic investigation".

To solve the problem "server-aided confidential forensic investigation", Hou et al. [1,2] presented several solutions under the assumption that the server administrator is willing to cooperate with the investigator to search the relevant data. The detail of their solutions is as follows: (1) the investigator specifies a single keyword or multiple keywords based on the investigation subject, encrypts it or them 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 capturing 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 is supposed to have responsibility for protecting the irrelevant data against disclosing, and at the same time he is prevented from learning the relevant data due to some special crimes. But without learning what the relevant data is, the administrator cannot judge if the relevant data is really *relevant* to the crimes and neither can check if the investigator obtained other irrelevant data from the server. Regarding this problem, the work [1,2] 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, even if the assumption works, no measures can guarantee that the presented data is the one that comes from the server and no alteration occurs to it. In other words, the authenticity and integrity of the evidence collected in the work [1,2] 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. Therefore, we put our major concern on how to prove the authenticity and integrity of the evidence collected in the work [1,2].

In this paper, we propose a "encryption-then-blind signature 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, we aim to realize that the administrator (or the third party the administrator trusts) can verify whether the presented evidence is the data that comes from the server and whether the evidence is altered or not. In addition, we implement the proposed system based on commutative encryption and examine its security.

2 Preliminaries: Commutative Encryption

In our proposed scheme, commutative encryption plays an important role.

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

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 is commutative for keys with a common modulus n. It was introduced by Rivest, Shamir and Adleman in 1978 ([3]) and one description of the system is described 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 $\mathcal{E}_k(\cdot)$ to denote the encryption operation with key k, it is obvious that

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

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

3 Proposed Scheme: Encryption-Then-Blind Signature with Designated Verifier

3.1 Requirements of "Server-Aided Confidential Forensic Investigation"

For clarity, we summarize the requirements of "server-aided confidential forensic investigation" below.

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

- 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 **administrator's** viewpoint, he hopes to fulfill the following:

- 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 open or presented as evidence in court. That is, the administrator needs to protect user data against unauthorized disclosing. If some data has to be open, it should be open in a secure manner.

3.2 Details of Proposed Scheme

For the brevity of description, we take single keyword case as an example and adopt the following notation. The single keyword specified by the investigator is denoted 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 $\mathcal{E}(w^*)$ and $\mathcal{E}(W)=\{\mathcal{E}(w_1), \mathcal{E}(w_2), \ldots, \mathcal{E}(w_v)\}$, where $\mathcal{E}(\cdot)$ is the encryption function.

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

1. **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 $\mathcal{E}_{p_I}(w^*)$ 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 $\{\mathcal{E}_{p_I}(W^1), \mathcal{E}_{p_I}(W^2), \ldots, \mathcal{E}_{p_I}(W^L)\}$. Both the keyword and the documents are encrypted, which are assumed to be provably secure in the sense that the administrator cannot learn anything about the specified keyword as it is encrypted and the investigator cannot learn more than the searching results. The searching results must contain the specified keyword, so the investigator can treat them as potential evidence.

2. **Blind signature** for authenticity and integrity
   On server side, the administrator performs the following.
   - uses $\mathcal{E}_{p_I}(w^*)$ to search over all the encrypted documents $\{\mathcal{E}_{p_I}(W^1), \mathcal{E}_{p_I}(W^2), \ldots, \mathcal{E}_{p_I}(W^L)\}$, and retrieves $\mathcal{E}_{p_I}(W)$ such that the plaintext document W contains the keyword w^* (i.e., $W \ni w^*$).
     There are several ways to judge whether the plaintext document W contains the keyword w^* based on the relation between the ciphertext $\mathcal{E}(W)$ and $\mathcal{E}(w^*)$. As $\mathcal{E}(\cdot)$ is a deterministic encryption (e.g., RSA cryptosystem), we can get $W \ni w^*$ if $\mathcal{E}(W) \ni \mathcal{E}(w^*)$, i.e., there exist one word $w_i \in W$ such that $\mathcal{E}(w_i) = \mathcal{E}(w^*)$; as $\mathcal{E}(\cdot)$ is a probabilistic encryption, $W \ni w^*$ can be shown by applying techniques like zero-knowledge proof (please refer to the work [1] where Paillier cryptosystem is used).
   - signs W blindly by computing $\mathcal{E}_A(\mathcal{E}_{p_I}(W))$ if $W \ni w^*$ and sends the investigator $\mathcal{E}_{p_I}(W)$ as well as $\mathcal{E}_A(\mathcal{E}_{p_I}(W))$, where $\mathcal{E}_A(\cdot)$ is commutative encryption with $\mathcal{E}_{p_I}(\cdot)$ and the subscript 'A' means that it is the administrator's encryption function. As $\mathcal{E}_A(\cdot)$ is public key encryption, $\mathcal{E}_A(\mathcal{E}_{p_I}(W))$ means encrypting $\mathcal{E}_{p_I}(W)$ with the public key of the administrator. As $\mathcal{E}_A(\cdot)$ is secret key encryption, the secret key is only known to the administrator. In the following, we consider that $\mathcal{E}_A(\cdot)$ 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.
     The signature has the following properties: **(a) Selective signature**: the administrator signs only the relevant data W ($W \ni w^*$) instead of all the data stored on the server for less computational cost; **(b) Blind signature**: the administrator wants to verify the authenticity and integrity of the original relevant data W ($W \ni w^*$) rather than its illegible encrypted form $\mathcal{E}_{p_I}(W)$, so he needs to sign W blindly, that is, sign W without knowing what the W is. Here, the administrator signs W blindly by computing $\mathcal{E}_A(\mathcal{E}_{p_I}(W))$, i.e., computing encryption of W twice; **(c) Designated verifier signature**: 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 signature verification. On the other hand, the investigator also needs the administrator's cooperation to prove that the relevant data does come from the server and no alteration occurs to it when it is presented as evidence. In a word, a designated verifier signature rather than public verified signature is required here. In our

scheme, only the administrator knows signing key and verification key, so only the administrator can verify the signature. The administrator can also delegate the verification key to the third party he trusts and let the third party verify the signature.

3. **Decryption**
   The investigator decrypts $\mathcal{E}_{p_I}(W)$ ($W \ni w^*$) with his private key and performs investigation on W for capturing evidence. He also decrypts $\mathcal{E}_A(\mathcal{E}_{p_I}(W))$(which is $\mathcal{E}_{p_I}(\mathcal{E}_A(W))$ as $\mathcal{E}_{p_I}(\cdot)$ and $\mathcal{E}_A(\cdot)$ are commutative) for obtaining the signed W, i.e., $\mathcal{E}_A(W)$. He keeps the $\mathcal{E}_A(W)$ for the later signature verification.
4. **Signature verification**
   When the W is presented as evidence in court, the administrator (or the third party the administrator trusts) verifies the signature by test if $\mathcal{D}_A(\mathcal{E}_A(W))=W$ is true, where $\mathcal{D}_A(\cdot)$ is the inverse of encryption process $\mathcal{E}_A(\cdot)$. In other words, the administrator (or the third party the administrator trusts) verifies if the evidence is the W that comes from the server and if it is altered or not. At the same time, this also helps the investigator to show the authenticity and integrity of the evidence.

4 Conclusions

The scheme proposed in the paper can be shown to satisfy all security requirements. We also implemented our scheme based on an RSA cryptosystem. The results show that the performance is acceptable. Due to the space limitation, both the details of the security analysis and the experimental results will be given in the full paper. For future work, we will consider multi-dimensional search (e.g., range search, equality search, etc.) over encrypted data to overcome the restriction of the keyword search.

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. Hou, S., Uehara, T., Yiu, S.M., Hui, L.C.K., Chow, K.P.: Privacy Preserving Confidential Forensic Investigation for Shared or Remote Servers. In: 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 378–383 (2011)
2. Hou, S., Uehara, T., Yiu, S.M., Hui, L.C.K., Chow, K.P.: Privacy Preserving Multiple Keyword Search for Confidential Investigation of Remote Forensics. In: 2011 Third International Conference on Multimedia Information Networking and Security, pp. 595–599 (2011)
3. Rivest, R.L., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 2(21), 120–126 (1978)