A Privacy-Preserving Approach for Collecting Evidence in Forensic Investigation

Shuhui Hou[*], Siu-Ming Yiu[†], Tetsutaro Uehara[‡] and Ryoichi Sasaki[§]
[*]Dept. of Information and Computing Science
University of Science and Technology Beijing, Beijing, China
Email: shuhui@ustb.edu.cn
[†]Dept. of Computer Science
The University of Hong Kong, Hong Kong, China
Email: smyiu@cs.hku.hk
[‡]Research Institute of Information Security
Wakayama, Japan
Email: uehara@riis.or.jp
[§]Graduate School of Science and Technology for Future Life
Tokyo Denki University, Tokyo, Japan
Email: sasaki@im.dendai.ac.jp

ABSTRACT

Capturing digital evidence is crucial for counteracting against computer and cyber crimes. The technique of cloning the whole harddisk (for single PC) for investigation is not feasible in large sharing systems (e.g. in a third-party email server, data center or cloud system). Privay is also a major concern as most of the data in these systems is not relevant to the crime case. The problem is how to retrieve the relevant information without the investigator knowing other irrelevant data while the server administrator does not know what the investigator is searching. To solve this problem, Hou et al. modelled the problem as a secure keyword searching problem and proposed a number of encryption-based schemes. While the schemes are theoretically sound, the efficiency is a concern. Besides, there are several shortcomings in their schemes. Data integrity and authenticity are not considered; re-encryption for each investigator is needed if there are multiple investigators. In this paper, we solve the same problem using the technique of secret sharing to improve efficiency. By exploiting the homomorphic property of the secret sharing schemes, data integrity and authenticity can be guaranteed using digital signature. Our solution can also handle multiple investigators more efficiently. We showed that our solution is more efficient by experiments and comparing the number of operations required by our solution with some existing work.

KEYWORDS

Confidential forensic investigation, Chinese Remainder Theorem, secret sharing, homomorphism property, third-party neutral

1. INTRODUCTION

Computer-related and cyber crimes are becoming rampant and caused a lot of damages to business, the public and the governments. Collecting digital evidence from storage device(s) is a crucial step in the investigation phase. However, with the development of computing technology (e.g., cloud system), evidence collection techniques developed for single PC or small-scale systems such as cloning the harddisks become infeasible as potential evidence may be distributed on a large number of servers. Moreover, the privay issue is another major concern. The servers under investigation may store data from thousands or even more irrelevant users. The investigator may have no rights to access the irrelevant data, in particular, some of them may involve confidential information. A trivial solution is to ask the server administrator to retrieve only the data relevant to the crime (or the suspect) and hand this to the investigator. This simple solution also may not work since the investigator may not want the administrator to know what he is looking for due to specific nature of the crimes. This motivates us to study the problem of how to efficiently retrieve relevant data (i,e., evidential data) from a huge amount of data while preserving the privacy of irrelevant users.

In this paper, we assume that the server administrator is willing to cooperate, search the relevant data, and return all located data to the investigator whenever a warrant is provided. We aim at solutions that can satisfy the following **requirements**.

- **(**1**)** Privacy - The investigator cannot learn anything related to the irrelevant data stored in the devices.
- **(**2**)** Confidentiality - The server administrator who performs the search does not know what the investigator is looking for.
- **(**3**)** Integrity and authenticity - The integrity and authenticity of the data need to be guaranteed.
- **(**4**)** Efficiency - The solutions must be reasonably fast.
- **(**5**)** Multiple investigators - There are more and more cases that involve multiple investigators. The solution must be extendable to handle multiple investigators efficiently.

Hou et al. [1], [2] is the first to abstract this problem as a secure keyword searching problem and provides solutions to tackle the problem. The idea behind their solutions is as follows. The investigator specifies single or multiple keyword(s) based on investigation subject, encrypts and sends it (or them) to the server administrator; the administrator encrypts all the data files stored on the server (where each data file is represented as a set of words), searches for encrypted keyword(s) in the encrypted data files and returns the relevant data (i.e., the data files containing the keyword(s)) to the investigator. By searching the encrypted data files with encrypted keyword(s), the administrator has no idea of what keyword(s) the investigator is looking for; by performing investigation only on the relevant data files returned by the administrator, the investigator has no idea of other irrelevant data files (i.e., the data files without containing the keyword(s)). It should be pointed out that while not perfect, keyword searching is currently the most widely recognized culling method in the area of digital forensics and e-discovery.

The schemes for single keyword search on encrypted data [1] are based on homomorphic encryption and commutative encryption, and the schemes for multiple keyword search [2] are based on the protocol for privacy preserving set intersection. These schemes can satisfy **Requirement (**1**)** and **(**2**)**. However, these schemes utilize encryption technology directly or indirectly, so the efficiency may be a concern (**Requirement (**4**)**) due to the time consuming encryption and decryption procedures on large amount of data. Moreover, these schemes do not satisfy **Requirement (**3**)** and **Requirement (**3**)** is important as if the integrity/authenticity of the data is questionable, it is not admissible to courts. For **Requirement (**5**)**, these existing schemes can only handle multiple investigators by encrypting the data for each investigator.

To tackle this problem, we propose to make use of (t, n)-threshold secret sharing schemes and their homomorphism properties to improve the investigation efficiency and verify the data integrity and authenticity. The high level idea is as follows. The data files managed by the server administrator are treated as a sequence of words. Each word and also each keyword given by the investigators are treated as secrets and are divided into n pieces of secret shares (or shadows). We employ a third-party neutral (e.g., technology experts) to match each word in a file with each keyword from the investigator, more precisely, the third-party neutral is matching the shares of each word to the shares of each keyword provided by the investigator. Once the shares of one word in a file matches the shares of a keyword, t shares of all remaining words of the same file as well as their signatures (generated by adding up some noises to them) will be forwarded to the investigator to reconstruct the whole file as well as its signature based on the principle of (t, n)-threshold secret sharing schemes. Note that the third-party neutral cannot learn anything about the keywords and the data files since he only knows the shares, not the original words. Its involvement in the matter is neutral and unbiased.

The reasons why we utilize (t, n)-threshold secret sharing schemes to improve confidential forensic investigation include: (i) in the cloud computing environment, there have been several data management services which are using secret sharing technology for managing/storing server data more securely (e.g., SecureCube/Secret Sharing of NRI SecureTechnologies, Ltd., Japan). It is reasonable to let the server administrator divide the data files into n pieces for protecting them from leaking; (ii) the secret sharing technology can divide data files into separate files and store them in different physical locations. Each separate file becomes meaningless and the original data files cannot be reconstructed from any *one* separate file. The original data files can be reconstructed even if *some* of the separate files are unobtainable as long as enough shares are obtained; (iii) the secret sharing schemes have homomorphism properties [3] which can be used to construct digital signatures, so that we can verify the data integrity and authenticity. It should be pointed out the main contribution of the paper is not developing new cryptographic schemes, but making use of well-known schemes to solve a real application. We will illustrate the practicality of our proposed solution using experiments and show that our solution is much faster than the schemes given in [1], [2].

The rest of the paper is organized as follows. In Section 2, we introduce (t, n)-threshold secret sharing scheme and its homomorphism property which are necessary for understanding our solution. Section 3 addresses how our solution works based on the secret sharing schemes. The performance evaluation is conducted and experimental results are analyzed in Section 4. In Section 5, we clarify that our solution can handle multiple investigators efficiently and we conclude the paper and highlight some of the future work in Section 6.

2. THRESHOLD SECRET SHARING SCHEMES

Definition 1. *Let $\mathcal{F}$ be a field. A (t, n)-threshold secret sharing scheme is a secret sharing scheme that can divide a secret $s \in \mathcal{F}$ into shares $\{s_1, s_2, \ldots, s_n\} \in \mathcal{F}$ so that $t \leq n$ and:*

1) *Given any set of t or more shares s_i, s can be reconstructed;*
2) *Any set of fewer than t shares gives no information about s.*

Secret sharing was introduced by Shamir and Blakley. These cryptographic schemes were first designed for key safeguarding, but have been applied far beyond its original intent. There exists a multitude of secret sharing schemes: Shamir's secret sharing scheme [4] uses curves and reconstructs the secret by polynomial interpolation; Blakley's scheme [5] reconstructs the secret by the intersection of hyperplanes; the Asmuth & Bloom secret sharing scheme [6] uses congruence classes to solve the secret sharing problem; the Mignotte's threshold secret sharing scheme [7] uses the Chinese Remainder Theorem to solve the problem of secret sharing.

In this paper, we adopt the Mignott's threshold secret sharing scheme to divide the data into pieces so as to protect the investigation subject and irrelevant data from unauthorized disclosing.

2.1. Chinese Remainder Theorem(CRT)

Theorem 1. *Given a set of simultaneous congruences*

$$x \equiv a_i \ (mod\ n_i) \tag{1}$$

for $i = 1, 2, \ldots, r$ and where all $n_i's$ are pairwise relatively prime, the solution of the set of congruences is

$$x \equiv a_1 b_1 \frac{N}{n_1} + \cdots + a_r b_r \frac{N}{n_r} \ (mod\ N) \tag{2}$$

where $N = n_1 n_2 \cdots n_r$, and the b_i are determined from

$$b_i \frac{N}{n_i} \equiv 1 \ (mod\ n_i) \tag{3}$$

2.2. Mignotte's Threshold Secret Sharing Scheme

Besides the CRT, Mignotte's threshold secret sharing scheme uses special sequences of integers, referred to as Mignotte sequences.

Definition 2. *Let n, t be positive integers, $n \geq 2$ and $2 \leq t \leq n$. An (t, n)-Mignotte sequence is a sequence of pairwise coprime positive integers $p_1 < p_2 < \ldots < p_n$ such that*

$$\prod_{i=0}^{t-2} p_{n-i} < \prod_{i=1}^{t} p_i. \tag{4}$$

The Mignotte's threshold secret sharing scheme works as follows:

- **Initialization**
Choose a (t, n)-Mignotte sequence such that $\beta < s < \alpha$, where the secret s is chosen as a random integer, $\alpha = \prod_{i=1}^{t} p_i$ and $\beta = \prod_{i=0}^{t-2} p_{n-i}$;
- **Creating the shares**
The secret shares s_i are computed by $s_i \equiv s$ mod p_i, for all $1 \leq i \leq n$;
- **Reconstructing the secret**
Given t distinct shares $s_{i_1}, s_{i_2}, \ldots, s_{i_t}$, where

i_1,i_2, ..., i_t are t numbers arbitrarily taken from {1,2,...,n}. Based on the CRT, the secret s is reconstructed as the unique solution modulo p_{i_1},p_{i_2},...,p_{i_t} of the system

$$\begin{cases} x \equiv s_{i_1} \mod p_{i_1} \\ \vdots \\ x \equiv s_{i_t} \mod p_{i_t} \end{cases} \quad (5)$$

2.3. Homomorphism Property

In order to verify the integrity and authenticity of the data, we can apply digital signatures. However, since the data is now divided into shares, we need the following homomorphic property to enable us to sign on the shares and combine these signature shares into a complete signature. Mignotte's threshold secret sharing scheme satisfies this property.

Let $\mathcal{S}$ be the domain of possible secrets and $\mathcal{K}$ be the domain of secret shares. Every instance of a (t, n)-Mignotte's threshold secret sharing scheme determines a set of functions $\mathcal{H}_I$: $\mathcal{K}^t \rightarrow \mathcal{S}$ defined for each $I \subseteq \{1, 2, \ldots, n\}$ with $|I| = t$. These functions define the value of the secret s given any set of t values $s_{i_1}, \ldots, s_{i_t}$:

$$s = \mathcal{H}_I(s_{i_1}, \ldots, s_{i_t}) \quad (6)$$

where $I = \{i_1, \ldots, i_t\}$.

Let $+$ and $\oplus$ be modular addition on elements of the secret domain $\mathcal{S}$ and of the share domain $\mathcal{K}$, respectively. For all I, if

$$s = \mathcal{H}_I(s_{i_1}, \ldots, s_{i_t}) \quad (7)$$

and

$$s' = \mathcal{H}_I(s'_{i_1}, \ldots, s'_{i_t}) \quad (8)$$

it is easy to prove that

$$s + s' = \mathcal{H}_I(s_{i_1} \oplus s'_{i_1}, \ldots, s_{i_t} \oplus s'_{i_t}). \quad (9)$$

In other words, a (t, n)-Mignotte's threshold secret sharing scheme is $(+, \oplus)$-homomorphic.

Similarly, provided that $s = \mathcal{H}_I(s_{i_1}, \ldots, s_{i_t})$ for all I, then for any $\gamma > 0$ we have

$$\gamma s = \mathcal{H}_I(\gamma s_{i_1}, \ldots, \gamma s_{i_t}). \quad (10)$$

Based on such kind of homomorphism property, we can verify the validity of secret shares without revealing them.

3. CONFIDENTIAL FORENSIC INVESTIGATION BASED ON SECRET SHARING

3.1. Notations

Recall that given a shared server, relevant data (or evidential data) is stored together with the irrelevant data (some may involve confidential information or private information) on the server. Whenever a warrant is provided, we assume that the server administrator is willing to cooperate and will not hide some of the files from being searched.

For the brevity of description, we adopt the following notations. Based on the investigation intent or the investigation subject, the investigator will specify a set of keywords, denoted as w^*={w_1^*, w_2^*, ..., w_u^*}, where each keyword w_j^* (1≤j≤u) is assumed to be d-bit long; The data stored on the server is viewed as a set of documents, denoted as {W^1, W^2, ..., W^L}. Any one document W^l∈{W^1, W^2, ..., W^L} consists of a sequence of words, denoted as a matrix W^l=$[w_1^l\ w_2^l\ \ldots\ w_v^l]^{\mathrm{T}}$ where the "T" in the top right-hand corner means matrix transpose and every word is also assumed to be d-bit long. Although the keywords or words are of variable length, we can transform them into equal length by picking a fixed-size block like the work [8] where words that are too short or too long may be padded to a multiple of the block size with some pre-determined padding format. We also can use hash function to map the variable-length words into the fixed-length words.

3.2. Details of Proposed Solution

Take "single keyword search" as an example, we describe how to realize confidential forensic investigation based on secret sharing.

We employ a third-party neutral (e.g., technology experts) to carry out the matching and signing process. Utilizing third-party neutral will ensure safe handling of evidence since it is impartial and this impartiality presumptively aids in the evidence-finding process and administration of justice.

1) To manage or store the server data securely, the **administrator** uses (t, n)-threshold secret sharing schemes to divide the data into

n pieces so that any one piece of data is meaningless and the original data cannot be reconstructed from any one piece of data. In order to verify if the data is the one collected from the server or verify if the data is changed or not when the data is presented as evidence in a court, he generates some pseudo-random numbers like noise and uses them to produce signatures of the data. In detail,

- He chooses n pairwise coprime positive integers p_1,p_2,. . .,p_n to construct a (t, n)-Mignotte sequence such that $\beta < w_i^l < \alpha$, where w_i^l $(1 \leq i \leq v)$ denotes any word of the document $W^l \in \{W^1, W^2, \ldots, W^L\}$, $\alpha = \prod_{i=1}^{t} p_i$ and $\beta = \prod_{i=0}^{t-2} p_{n-i}$. As every word is d-bit long, we have $2^{d-1} \leq w_i^l < 2^d$.
- To prevent the investigator from knowing the irrelevant data, he views each word w_i^l of the document W^l as secret and divides it into n secret shares by computing

  $$w_{ik}^l \equiv w_i^l \bmod p_k, \tag{11}$$

  for all $1 \leq i \leq v$, $1 \leq k \leq n$. For convenience of description, we use $\left[W^l\right]_{\text{mod}}$ to denote the matrix consisting of secret shares of all words in the document W^l,

  $$\left[W^l\right]_{\text{mod}} = \begin{bmatrix} w_{11}^l & w_{12}^l & \cdots & w_{1n}^l \\ w_{21}^l & w_{22}^l & \cdots & w_{2n}^l \\ \cdots & \cdots & \cdots & \cdots \\ w_{v1}^l & w_{v2}^l & \cdots & w_{vn}^l \end{bmatrix}. \tag{12}$$

  That is, one row corresponds to one word of W^l.
- He generates a random number γ^l and a sequence of pseudo-random numbers denoted as a matrix $E^l = [e_1^l \ e_2^l \ \ldots \ e_v^l]^T$ for each document W^l. He also views each number e_i^l of E^l as secret and divides it into n secret shares by computing

  $$e_{ik}^l \equiv e_i^l \bmod p_k, \tag{13}$$

  for all $1 \leq i \leq v$, $1 \leq k \leq n$. We use $\left[E^l\right]_{\text{mod}}$ to denote the matrix consisting of secret shares of all numbers in E^l, that is,

  $$\left[E^l\right]_{\text{mod}} = \begin{bmatrix} e_{11}^l & e_{12}^l & \cdots & e_{1n}^l \\ e_{21}^l & e_{22}^l & \cdots & e_{2n}^l \\ \cdots & \cdots & \cdots & \cdots \\ e_{v1}^l & e_{v2}^l & \cdots & e_{vn}^l \end{bmatrix} \tag{14}$$

- As an investigator requests him to cooperate in investigation, he will send the n pairwise coprime positive integers p_1,p_2,. . .,p_n to the investigator, and provides third-party neutral (e.g., technology experts) $\left[W^l\right]_{\text{mod}}$ as well as its correspoinding $\left[E^l\right]_{\text{mod}}$, γ^l.

2) For preventing the administrator and the third-party neutral from knowing the investigation subject, the **investigator** specifies one keyword w_j^*, views it as secret and divides it into n secret shares by computing

$$w_{jk}^* \equiv w_j^* \bmod p_k, \tag{15}$$

for all $1 \leq k \leq n$; He randomly picks t distinct shares $w_{jk_1}^*$,. . .,$w_{jk_t}^*$ of w_j^* (k_1,k_2, . . ., k_t are t numbers from $\{1$,2,. . .,$n\}$) and gives them to the third-party neutral.

3) Without knowing n pairwise coprime positive integers p_1,p_2,. . .,p_n, the **third-party neutral** uses t secret shares $w_{jk_1}^*$,. . .,$w_{jk_t}^*$ to match the matrix $\left[W^l\right]_{\text{mod}}$ row by row. If there exists i-th row $(1 \leq i \leq v)$ whose ik_1-,. . .,ik_t-entries are equal to $w_{jk_1}^*$,. . .,$w_{jk_t}^*$ (which indicates W^l contains the keyword w_j^*), the **third-party neutral** will pick the t columns of the matrix $\left[W^l\right]_{\text{mod}}$ denoted by

$$\left[W^l\right]_{v \times t} = \begin{bmatrix} w_{1k_1}^l & w_{1k_2}^l & \cdots & w_{1k_t}^l \\ w_{2k_1}^l & w_{2k_2}^l & \cdots & w_{2k_t}^l \\ \cdots & \cdots & \cdots & \cdots \\ w_{vk_1}^l & w_{vk_2}^l & \cdots & w_{vk_t}^l \end{bmatrix} \tag{16}$$

and computes

$$\left[S^l\right]_{v \times t} = \gamma^l \left[W^l\right]_{v \times t} + \left[E^l\right]_{v \times t} \tag{17}$$

where $\left[E^l\right]_{v \times t}$ stands for the correspoinding t columns of the matix $\left[E^l\right]_{\text{mod}}$. Then he returns the investigator $\left[W^l\right]_{v \times t}$ as well as $\left[S^l\right]_{v \times t}$. The **third-party neutral** cannot reconstruct the keywords and the documents without knowing n pairwise coprime positive integers, i.e., he will learn nothing about the keywords and the documents. Furthermore, the third-party neutral separates the administrator from searching results so that the administrator has no idea of what the investigator is looking for, and the third-party neutral only returns the documents containing the matched keywords so that the investigator cannot know other irrelevant documents.

4) Based on the matrix $\left[W^l\right]_{v\times t}$ and $\left[S^l\right]_{v\times t}$ which consists of t secret shares of each word and signed word of W^l, the **investigator** can reconstruct each word and its signature, further can reconstruct the whole document W^l and its signature S^l. Consequently, he can perform investigation on W^l for capturing evidence. With the cooperation from the administrator (he provides E^l and γ^l), the investigator can prove the integrity and authenticity of the W^l by verifying if

$$\gamma^l W^l = S^l - E^l. \tag{18}$$

From the equation (17), we have

$$\begin{cases} s^l_{ik_1} & \equiv & \gamma^l w^l_{ik_1} + e^l_{ik_1} \mod p_{k_1} \\ & \vdots & \\ s^l_{ik_t} & \equiv & \gamma^l w^l_{ik_t} + e^l_{ik_t} \mod p_{k_t} \end{cases} \tag{19}$$

for all $1\leq i\leq v$. As $w^l_i = \mathcal{H}_I(w^l_{ik_1},\ldots,w^l_{ik_t})$ and $e^l_i = \mathcal{H}_I(e^l_{ik_1},\ldots,e^l_{ik_t})$, from the homomorphism property of $\mathcal{H}_I$ it follows that

$$\begin{aligned} s^l_i &= \mathcal{H}_I(s^l_{ik_1},\ldots,s^l_{ik_t}) \\ &= \mathcal{H}_I(\gamma^l w^l_{ik_1} \oplus e^l_{ik_1},\ldots,\gamma^l w^l_{ik_t} \oplus e^l_{ik_t}) \\ &= \gamma^l \mathcal{H}_I(w^l_{ik_1},\ldots,w^l_{ik_t}) + \mathcal{H}_I(e^l_{ik_1},\ldots,e^l_{ik_t}) \\ &= \gamma^l w^l_i + e^l_i, \end{aligned} \tag{20}$$

for all $1\leq i\leq v$. Thus, we have

$$S^l = \gamma^l W^l + E^l. \tag{21}$$

So we can verify the signature by checking if the equation (18) is true or not.

Based on the “single keyword search”, “multiple keyword search” can be easily realized. The investigator provides the third-party neutral t secret shares of w^* (i.e., $w^*_{jk_1}$,. . .,$w^*_{jk_t}$, $1\leq j\leq u$) and the third-party neutral matches them with the secret shares of the document W^l ($1\leq l\leq L$). The third-party neutral will return t secret shares of the document W^l which contains $w^*_{jk_1}$,. . .,$w^*_{jk_t}$, $1\leq j\leq u$. Then, the investigator can reconstruct each word and the whole document which contains multiple keywords. The signature signing and verification process can be performed similarly.

4. PERFORMANCE EVALUATION

4.1. Security Analysis

In this section, we show that our proposed solution satisfies the security **requirements**. First, the keywords to be searched are divided into shares, the server administrator has no way to deduce what keywords or what subjects the investigator want to search. In other words, the confidentiality of the investigation subject (or keywords specified by the investigator) is protected (**Requirement (2)**). On the other hand, only data that can satisfy the search criteria will be passed to the investigator, thus the privacy of irrelevant data is preserved (**Requirement (1)**). Also, without knowing the n pairwise coprime positive integers p_1,p_2,. . .,p_n, the third-party neutral cannot reconstruct the specified keywords and the server data even he knows all the shares of data. Thus, no information leak occurs on the third-party neutral side.

In addition, the third-party neutral signs the meaningless shares of words of relevant data W^l by adding some meaningless shares of E^l. He cannot get any meaningful information without knowing the n pairwise coprime positive integers. In our solution, only the administrator can verify the signature so that he can check if the presented evidence does come from the server and if it is altered or not. In other words, he can check if the server data is used in a secure way by signature verification. This fact can also helps the investigator to prove the authenticity and integrity of presented evidence so that it can be admitted in a court (**Requirement (3)**).

4.2. Computational Complexity

For the sake of simplicity, we merely evaluate the computational complexity as one keyword w^*_j is input and one document W^l containing w_j^* is output. We use the number of modular operations (MO), modular multiplications (MM), modular exponentiations (ME) and modular inversions (MI) to measure the computational complexity.

In our solution, the investigator needs n MO to compute the shares of keyword w_j^* and needs tv MI and tv MM to reconstruct the document W^l; the administrator needs nv MO to compute the shares of the document W^l; the third-party neutral needs t^2v comparison operations to perform searching and returning the relevant data W^l to the investigator.

To show the advantage of our solution, we also analyze the computational complexity of one existing work [1] which is based on Paillier cryptosystem. The investigator needs 2 ME and 1 MM to encrypt the keyword w_j^* and needs v ME, v MI and v MM to decrypt the document W^l; the administrator needs $2v$ ME and v MM to encrypt the the document W^l (we omit operations involved in the zero knowledge proof for the clarity).

We listed the above computational complexity in Table 1. ME and MI are usually more complex than MO and MM, so our solution is superior over the existing work in terms of computational complexity (**Requirement** (4)).

4.3. Experiments Evaluation

We base on Paillier cryptosystem as an example. We evaluate the efficiency of our proposed solution by experiments. We show that our solution has faster processing time than the existing work. We conducted the experiments on a Genuine Intel(R) CPU U7300, 1.30 GHz PC with 2 GB RAM, MATLAB 7 as the integrated environment. We take a word document (consisting of 273 English words separated by spaces), randomly set the positions where the keyword appears five times, and use the average processing time to measure the efficiency. We set the following parameters in our experiment:

1) Proposed solution based on secret sharing
Let $t = 3$, $n = 5$, $p_1 = 5$,$p_2 = 7$,$p_3 = 11$,$p_4 = 13$,$p_5 = 17$, then $\alpha = \prod_{i=1}^{t} p_i$=$385$ and $\beta = \prod_{i=0}^{t-2} p_{n-i}$=$221$; The "preprocessing" includes using MD5 hash function to transform the variable-length words to fixed-length words so that each word satisfies the condition of $(3, 5)$-threshold secret sharing.
2) Existing work based on Paillier cryptosystem
The "preprocessing" includes using MD5 hash function to transform the variable-length words to fixed-length words so that each word satisfies the condition of Paillier cryptosystem, which is detailed below.

- Key generation
Let $p = 3$,$q = 11$, then n=pq=33; Let $g = 166$, compute λ=$lcm(p-1,q-1)$=10 and μ=$(\Phi(g^\lambda \text{mod} \quad n^2))^{-1}$mod $\quad n$=2, where $\Phi(u) = \frac{u-1}{n}$. That is, the public key is (n, g)=$(33, 166)$ and the secret key is (λ, μ)=$(10, 2)$;
- Encryption
Let m (m<n and m∈Z_n) be plaintext, pick a random number r∈Z_n^*, the ciphertext c=$g^m r^n$ mod n^2;
- Decryption
The plaintext m=$\Phi(c^\lambda \text{mod } n^2) \cdot \mu$ mod n.

The experimental results are shown in Table 2, where dividing and reconstructing in our proposed solution are corresponding to the encryption and decryption procedures in existing work respectively, and "s" (i.e., "second") is the execution cputime. The processing time in our solution, especially the time on dividing the document into shares and searching are less than the processing time on encrypting the document and searching in existing work. Further, we carried out the above experiments on several larger size documents. As the document size is growing, the encryption time and searching time in existing work grow linearly while the time of secret dividing and searching time in our solution stay more or less the same (please refer to Figure 1). Thus, our solution is more scalable. Thus, our solution satisfies **Requirement (**4**)**.

On the other hand, dividing data into n shares may result in more storage cost than encrypting the data, but it may not be a problem as it is not necessary to store the shares of all files before performing the matching. Once the matching is done for a file, it is no longer needed.

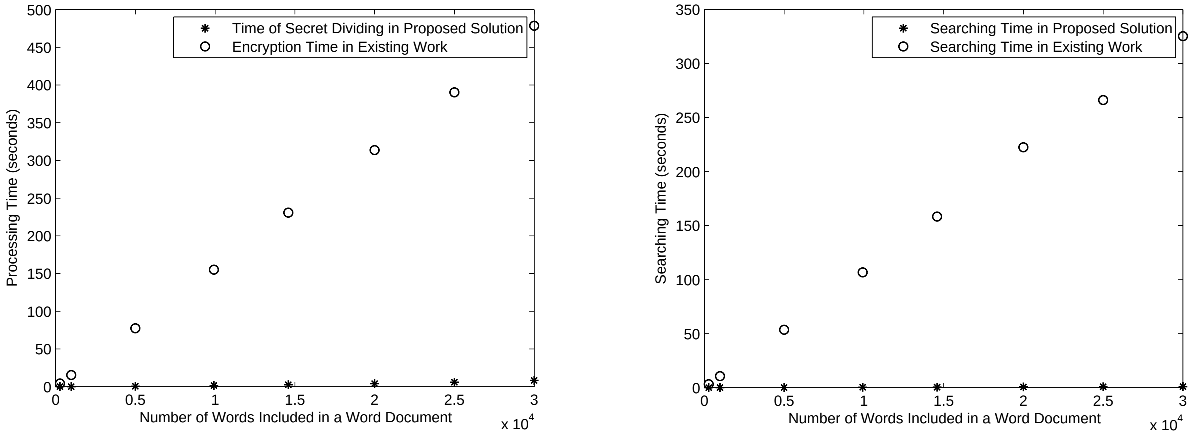
Table 1. Computational Complexity

	Investigator Side				Administrator Side		
	MO	MM	ME	MI	MO	MM	ME
Proposed Solution	n	tv	$-$	tv	nv	$-$	$-$
Existing Work	$-$	$1+v$	$2+v$	v	$-$	v	$2v$

"-": such operations are not required or the computational complexity is negligible.

Table 2. Processing Time

	Preprocessing	Dividing/ Encryption	Reconstructing/ Decryption	Searching
Proposed Solution	1.6419 s	0.0351 s	0.1664 s	0.0312 s
Existing Work	1.6536 s	4.3719 s	2.3868 s	3.2604 s



(a)Secret Dividing/Encryption time

(b)Searching Time

Figure 1. Processing Time on Larger Size Documents

5. MULTIPLE INVESTIGATORS

Another advantage of our solution is to support multiple investigators efficiently (**Requirement (**5**)**). Generally, prosecution and defense attorneys are supposed to be opposite parties in the criminal justice process. The prosecution attorney aims at getting a conviction and the defense attorney aims at finding facts that prove innocence. They stand in opposite positions so they have different requests during the procedure of investigation. In other words, there exist some cases where multiple investigators with different investigation intent need to perform investigation on the same data set.

Existing work [1], [2] cannot support multiple investigators *efficiently*, where the administrator has to re-encrypt the entire server data for each investigator with a different encryption scheme. In our proposed solution, it is easy to support multiple investigators. The administrator can cooperate with multiple investigators by sending the n pairwise coprime positive integers p_1,p_2,. . .,p_n of (t, n)-Mignotte sequence to each of them (e.g., prosecution and defense attorneys). The investigators can choose their own keywords to be sent to the third-party neutral. Of course, the third-party neutral may be able to discover that the same keyword is searched by different investigators. Since he does not know the exact keyword, this may not be a major concern.

6. CONCLUSIONS

To summarize, in this paper, we show that using a (t, n)-threshold secret sharing scheme, we can solve a real problem in computer forensic investigation.

The experimental results show that our solution is superior over the existing work in terms of computational complexity and practical performance. In practice, we can replace MD5 by SHA-256 in the solution if we want to increase the security level. This will affect the pre-processing time (it is estimated to be about three times longer) but the other processing time stays the same. Besides, we explore the homomorphism property of the (t, n)-threshold secret sharing scheme to realize signature signing and signature verification, which help the investigator to prove the data authenticity and integrity of the presented evidence so that it can be admitted in a court.

For future work, this paper and also all previous work do not provide a total solution to solve the real forensic investigation problem yet. For example, in these schemes, the administrator has no way to make sure that the keywords provided by the investigator are all relevant to the crime case. And the schemes also cannot guarantee that all files kept by the administrator can be searched over. All these require more investigation. In this paper, we pick one of the threshold secret sharing schemes as illustration, other schemes may work equally well or even better. A better and specially designed solution for solving this real application is always desired. We are now in the process of applying this result to some real crime cases and the feasibility and efficiency of the solution in real cases will be further investigated.

ACKNOWLEDGMENT

This work is partially supported by “Heiwa Nakajima Foundation, Japan”, partially sponsored by “the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry” and partially sponsored by “the National Science Foundation of China (No.11201025, No.11101028)”.

A preliminary version of this paper was presented at the International Conference on Cyber Security, Cyber Peacefare and Digital Forensics (CyberSec2013), March 2013.

REFERENCES

1 S. Hou, T. Uehara, S.M. Yiu, Lucas C.K. Hui, K.P. Chow: 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 S. Hou, T. Uehara, S.M. Yiu, Lucas C.K. Hui, K.P. Chow: 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 J. Benaloh: Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret. In: Advances in Cryptology-CRYPTO’86, LNCS 263, pp.251-260 (1987).

4 A. Shamir: How to share a secret. Commun. ACM, vol.22, No.11, pp.612-613 (1979).

5 G.R. Blakley: Safeguarding cryptographic keys. In: AFIPS Conference Proceedings, vol.48, pp.313-317 (1979).

6 C. Asmuth, J. Bloom: A modular approach to key safeguarding. IEEE Transactions on Information Theory, vol.29, No.2, pp.208-210 (1983).

7 M. Mignotte: How to share a secret?. In: Cryptography - Proceedings of the Workshop on Cryptography, Lecture Notes in Computer Science, vol.149, pp.371-375 (1983).

8 D. Song, D. Wagner, A. Perrig: Practical Techniques for Searches on Encrypted Data, In: Proceedings of IEEE Symposium on Security and Privacy 2000, pp.44-55(2000).