Title	Privacy preserving confidential forensic investigation for shared or remote servers
Author(s)	Hou, S; Uehara, T; Yiu, SM; Hui, CK; Chow, KP
Citation	The 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2011), Dalian, China, 14-16 October 2011. In Proceedings of the 7th IIHMSP, 2011, p. 378-383
Issued Date	2011
URL	http://hdl.handle.net/10722/152021
Rights	International Conference on Intelligent Information Hiding and Multimedia Signal Processing proceedings. Copyright © IEEE, Computer Society.

2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Privacy Preserving Confidential Forensic Investigation for Shared or Remote Servers

Shuhui Hou
Dept. of Information and Computer Science
University of Science and Technology Beijing
Beijing, China
Email: shuhui@ustb.edu.cn

Tetsutaro Uehara
Academic Center for Computing and Media Studies
Kyoto University
Kyoto, Japan
Email: uehara@media.kyoto-u.ac.jp

S.M. Yiu
Dept. of Computer Science
The University of Hong Kong
Hong Kong
Email: smyiu@cs.hku.hk

Lucas C.K. Hui
Dept. of Computer Science
The University of Hong Kong
Hong Kong
Email: hui@cs.hku.hk

K.P. Chow
Dept. of Computer Science
The University of Hong Kong
Hong Kong
Email: chow@cs.hku.hk

***Abstract*—It is getting popular that customers make use of third party data service providers to store their data and emails. It is common to have a large server shared by many different users. This creates a big problem for forensic investigation. It may not be easy to clone a copy of data from the storage device(s) due to the huge volume of data. Even if it is possible to make a clone, there are many irrelevant information/data stored in the same device for which the investigators have no right to access. The other alternative is to let the service provider search the relevant information and retrieve the data for the investigator provided a warrant can be provided. However, sometimes, due to the confidentiality of the crime, the investigator may not want the service provider to know what information they are looking for or the service provider herself may be one of the suspects. The problem becomes even more obvious in terms of cloud computing technology. In this paper, we address this problem and using homomorphic encryption and commutative encryption, we provide two forensically sound schemes to solve the problem so that the investigators can obtain the necessary evidence while the privacy of other users can be protected and at the same time, the service provider cannot know what information the investigators are interested in.**

***Keywords*-privacy preserving forensics; search on encrypted data; homomorphic encryption; commutative encryption**

I. Introduction

In recent years, there is an increasing number of computer and cyber crimes. It becomes a serious problem for businesses, the public, and government. How to capture digital evidence is critical for counteracting against computer crimes. On the other hand, it is getting popular that users do not host the data themselves, but make use of a third data service provider to store their data and/or emails. It is common to have a large server shared by many different users. This increases the difficulty of forensic investigation. The problem becomes even more difficult if we are talking about the cloud technology since the data is stored in a distributed manner and may involve a large number of servers and storage devices. The storage devices may be remote as well. It is quite obvious that traditional forensic technique may not be applied easily. For example, it may be difficult to clone a "copy" of data from the storage device(s) due to the huge volume of data and the distributed manner of the storage device(s).

Even if it is feasible to make a clone, there are many irrelevant information/data stored in the same device for which the investigators have no right to access. This data may involve confidential information and private information. The other alternative is to let the service provider search the relevant information and retrieve the data for the investigator provided a warrant can be provided. However, sometimes, due to the confidentiality of the crime, the investigator may not want the service provider to know what information they are looking for or the service provider herself may be one of the suspects.

In this paper, we address this problem and using homomorphic encryption and commutative encryption, we provide two forensically sound schemes to solve the problem so that the investigators can obtain the necessary evidence while the privacy of other users can be protected and at the same time, the service provider cannot know what information the investigators are interested in. So far, there is no forensically sound solution to solve this problem.

We assume that the evidence required by the investigator is stored together with a huge amount of irrelevant data on a remote server or a distributed set of storage devices. It is not possible to make a clone of all data. The service provider is willing to cooperate and search the relevant information for the investigator. However, they want to make sure that only relevant information will be given to the investigator, no other information of other users will be disclosed to the investigator. At the same time, the investigator does not want the service provider to know what information they are searching. We further assume that the service provider is trustable in the sense that he will not hide any information if it satisfies the searching

*This work is partially supported by "the Fundamental Research Funds for the Central Universities" (06108041).

978-0-7695-4517-2/11 $26.00 © 2011 IEEE
DOI 10.1109/IIHMSP.2011.28

criteria of the investigator. In other words, the service provider will give out all the information located.

Our main idea is as follows. The server administrator will encrypt all the data stored on the server for preventing the investigator from learning the irrelevant data; the investigator will provide the administrator keywords (which are in an encrypted form for preventing the administrator from learning the investigation subject) and the "trapdoor" so that the administrator can search for the relevant data from the encrypted data; the administrator will only return the relevant data to the investigator and the investigator will only decrypt and perform investigation on such relevant data for capturing the evidence.

There are a number of studies on searching for data without revealing its content to the server (sometimes called the "database" that stores data), but the problems that they try to tackle are different from ours. We group the major existing work into private database search (database data belong to the data owner who wishes to retrieve data) and public database search (database data is public such as stock quotes and someone who is not data owner wishes to retrieve data).

1) **Private database search** includes two kinds of scenarios:
   - Searching on private-key-encrypted data (Shortly, SSKE, [1][2][3]). The problem of SSKE was raised for the first time by Song et al. [1]. In the setting of SSKE, a user wishes to store his private data to a remote server while preventing the untrusted server administrator from learning the data. One solution is that the user himself encrypts the data before storing. However, it is difficult for the administrator to help the user retrieve his data later as it is in an encrypted form. In order to efficiently retrieve or search on the encrypted data, the user can organize his data in an arbitrary way before encryption and attach additional data structures such as secure indexes [4], capability [5] and hash functions/hash tables [6] [7], etc. Such additional data structures are also encrypted and stored on the server alongside the encrypted data, which are helpful to improve searching efficiency since each of them is associated with the encrypted data.
   - Searching on public-key-encrypted data (Shortly, SPKE, [8][9][10][11]). In the setting of SPKE, the user wants to retrieve his e-mails containing a certain keyword from the mail server, where the e-mails are encrypted by the senders using his public key. Anyone with access to the public key can encrypt the email but only the owner of the private key can generate "trapdoors" to perform retrieval and decryption. Different from the setting of SSKE, the data is encrypted by the senders and collected by the mail server, so the user cannot organize the data in any convenient way. The additional data structures are also introduced in the SPKE ([10][11]) for improving the searching efficiency and security.
2) **Public database search** Private information retrieval (Shortly, PIR, [12][13]) schemes allow a user to retrieve records from a database without revealing what records were retrieved and with total communication less than the data size. The original PIR scheme [12] allows the user to retrieve a record of the database only by address, which was extended to keyword searching including searching on streaming data [14]. Unlike the above two settings, the data in PIR is always unencrypted and any scheme that tries to hide the access pattern must touch all data items. Otherwise, the database will learn access pattern, namely, that the untouched item was not of interest to the user. Thus, the user needs to download records that he is not interested in. Consequently, communication cost is increased.

Our problem does not fit either of the two settings mentioned above. Unlike the **private database search**, the investigator is neither data owner nor receiver of data so he cannot manage the data in any convenient way. Moreover, he is only allowed to investigate the relevant data instead of the all for protecting the privacy of innocent data. Unlike the **public database search**, keywords derived from the investigation subject and server data need to be in an encrypted form for protecting the privacy on both sides: investigation subject and innocent data. We summarize the difference between our work and the existing work in TABLE I.

In this paper, based on the homomorphic encryption and commutative encryption schemes, we will present two forensically sound proposals to assist the investigator in searching for evidence efficiently without exposing irrelevant data to the investigator. The remainder of the paper is organized as follows. In Section II, we make assumptions to formulate our problem and clarify its requirements. Section III presents the first proposal based on homomorphic encryption. The second one which is based on commutative encryption is stated in Section IV. Finally, discussions are conducted and conclusions are drawn in Section V.

II. Problem Formulation

We make the following assumptions.

1) The investigator and the administrator do not trust each other. To prevent the administrator (who may be a potential suspect) from learning the investigation subject, the investigator will provide the administrator keywords which are in an encrypted form. To prevent the investigator from obtaining innocent data, the administrator will verify what keywords are used later. For example, during the evidence presentation in a court of law, the investigator can be required to show what evidence is collected based on what keywords, so the administrator can check whether the investigator cheated for obtaining other information from the server.
2) Evidential data is stored alongside the innocent data on a remote server in non-encrypted form. For simplicity, we view the data as a set of documents and each document

	Problem Setting	Procedures
SSKE	data owner wishes to outsource his private data to (untrusted) server	data owner encrypts the data and server performs the search on encrypted data
SPKE	receiver of emails wishes to manage his emails on (untrusted) server	senders of emails encrypt emails and server performs the search on encrypted emails
PIR	user wishes to query database without revealing query and results	user queries unencrypted database data using oblivious transfer technologies, etc.
Our Work	investigator can efficiently capture evidence without revealing innocent data	server administrator encrypts the data and performs the search on encrypted data

TABLE I

W is a series of word blocks which has fixed length as follows:

$\cdots$	w_{i-1}	w_i	w_{i+1}	$\cdots$

We assume that each keyword specified by the investigator has the same length as w_i.

3) It is difficult to distinguish the relevant data from the irrelevant ones. We view the documents containing the specified keywords as relevant data and those without containing the specified keywords as irrelevant ones. Then, the investigator is only allowed to perform investigation on documents which contain the specified keywords.
4) Both the keywords and the data stored on the server are encrypted by the cryptographic scheme, which is assumed to be provably secure in the sense that the server administrator cannot learn anything about the specified keywords when they are encrypted and the investigator cannot learn more than the search result. The search result must contain the specified keywords, so the investigator can treat them as potential evidence.

We formulate our problem in TABLE II.

A scheme which satisfies the following properties is desirable for our problem.

Inputs	*Investigator*	w^*: specified keyword
	Server administrator	D: whole set of server-side documents
Outputs	*Investigator*	Nothing
	Server administrator	$W(\in D)$: document involving w^*
Privacy	*Investigator*	Server administrator cannot learn w^*
	Server administrator	Investigator cannot learn more than W

TABLE II

To protect the privacy on both sides: investigator and server administrator, the keyword w^* and the documents D need to be encrypted. This will lead to a non-index, sequential search on the entire server. Besides, public key encryption is required. Both the investigator and the server administrator can perform encryption but only the one who owns private key can perform decryption.

III. A SCHEME BASED ON HOMOMORPHIC ENCRYPTION

After an event involving computer crime has occurred, the investigator or the police usually search for evidence over all the documents stored on the server. However, as the data is irrelevant to the crimes and contains confidential information or privacy information, data owners may be unwilling to reveal it to the investigator. Data owners usually trust the administrator who is responsible for managing the data in a secure manner. Hence, the alternative is to let the administrator perform the searching and only return the relevant data to the investigator. Take the company server as an example, if there are only a few employees suspected, the administrator usually provides the investigator their data rather than all the employees' data. Here, we assume that the administrator honestly returns all the searching results without holding some of them.

In our first scheme, the investigator encrypts the specified keyword w^* for preventing the administrator from learning the investigation subject and the administrator encrypts the whole set of documents D with the public key of the investigator for the sake of searching; the administrator uses the encrypted keyword w^* to search on the encrypted set D and only returns the relevant encrypted document W to the investigator. As shown on Fig. 1, the upper half part is what the administrator performs while the lower half part is what the investigator performs. As a result, the investigator can avoid investigating what he is not interested in and the privacy of innocent data can also be protected from revealing. Besides, neither the keywords nor the search results can be decrypted by the server administrator since he does not know the private key. We realize "searching on encrypted data with encrypted keyword" by utilizing the homomorphic encryption, which can help perform searching on ciphertext.

A. Homomorphic Encryption

The most common definition of homomorphic encryption ([15]) is the following.

Definition 1: Let $\mathcal{M}$ (resp., $\mathcal{C}$) denote the set of plaintexts (resp., ciphertexts). An encryption scheme is said to be homomorphic if for any given encryption key k the encryption function E satisfies

$$\forall\ m_1, m_2 \in \mathcal{M}, E(m_1\ \odot_{\mathcal{M}}\ m_2) \leftarrow E(m_1) \odot_{\mathcal{C}} E(m_2) \quad (1)$$

for some operators $\odot_{\mathcal{M}}$ in $\mathcal{M}$ and $\odot_{\mathcal{C}}$ in $\mathcal{C}$, where "$\leftarrow$" means "can be directly computed from", that is, without any intermediate decryption.

From this definition it follows that, given a fixed key, performing operations $\odot_{\mathcal{M}}$ on the plaintexts before encryption is equivalent to performing operations $\odot_{\mathcal{C}}$ on the corresponding ciphertexts after encryption.

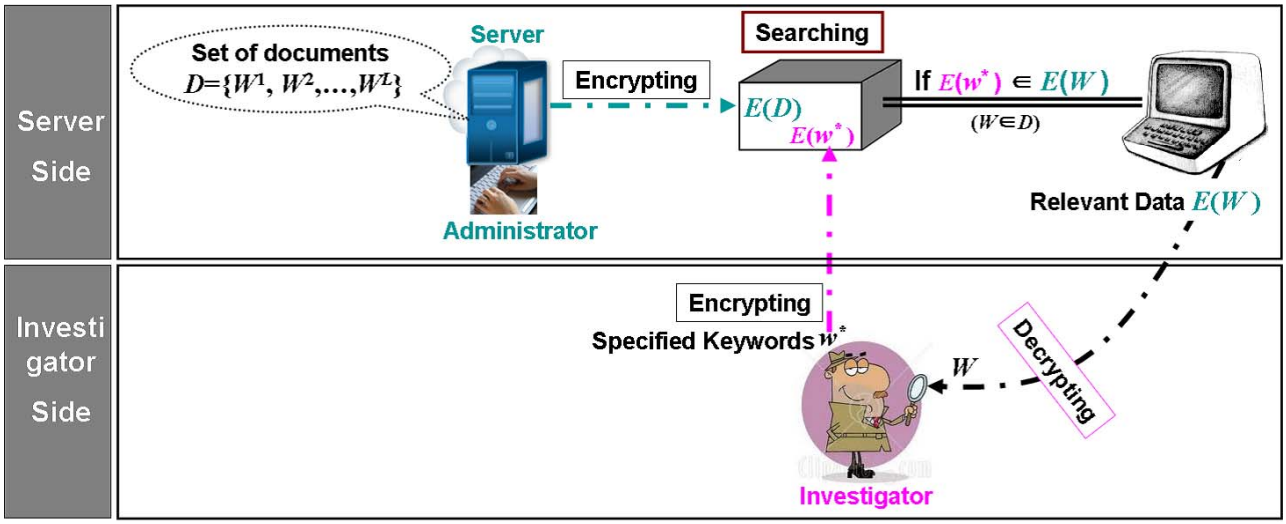

Fig. 1. Searching on Encrypted Data with Encrypted Keyword

Prerequisite: Alice computed a (public, private) key: she first chose an integer $n = pq$, p and q being two large prime numbers and n satisfying $\gcd(n, \phi(n)) = 1$, and considered the group $G = Z^*_{n^2}$ of order k. She also considered $g \in G$ of order n. Her public key is composed of n and g, and here private key consists in the factors of n.
Goal: Anyone can send a message to Alice.
Principle: To encrypt a message $m \in Z_n$, Bob picks at random an integer $r \in Z^*_n$, and computes $c = g^m r^n \bmod n^2$. To get back to the plaintext, Alice computes the discrete logarithm of $c^{\lambda(n)} \bmod n^2$, obtaining $m\lambda(n) \in Z_n$, where $\lambda(n)$ denotes the Carmichael function. Now, since $\gcd(\lambda(n), n) = 1$, Alice easily computes $\lambda(n)^{-1} \bmod n$ and gets m.

Fig. 2. Paillier cryptosystem in [15]

B. Paillier cryptosystem

The Paillier cryptosystem is named after and invented by Pascal Paillier in 1999 [16], which is a public key cryptography shown in Fig. 2. Without special remarks, we adopt the notations in [15] directly in the rest of paper.

The Paillier cryptosystem works based on the function $\varepsilon_g : Z_n \times Z^*_n \to Z^*_{n^2}$, which maps $(m, r) \to g^m \cdot r^n$. A notable feature of the Paillier cryptosystem is its additive homomorphic properties: $D(E(m_1) \cdot E(m_2) \ mod\ n^2) = m_1 + m_2 \ mod\ n$, where D is the decryption function. Our first scheme is based on the Paillier cryptosystem.

C. Details of Scheme

Suppose that the investigator inputs the keyword w^* and the administrator inputs the set of documents $D = \{W^1, W^2, \ldots, W^L\}$. With Paillier cryptosystem, the investigator encrypts the keyword w^* for preventing the administrator from learning the investigation subject by

$$E(w^*) = g^{w^*} \cdot r^n \ mod\ n^2 \tag{2}$$

where r is random number for semantic security. The administrator encrypts the each word block w_i of document W $(W \in D)$ by

$$E(w_i) = g^{w_i} \cdot r_i^n = g^{w_i} \ mod\ n^2 \tag{3}$$

where the random number r_i is taken as 1. It is reasonable to set r_i=1 since we assume that data owners trust the server administrator and the data is encrypted for preventing the investigator rather than the administrator from learning the innocent data. It follows that

$$\delta_i = \frac{E(w^*)}{E(w_i)} = \frac{g^{w^*} \cdot r^n}{g^{w_i}} = r^n \ mod\ n^2 \ only\ if\ w^* = w_i \tag{4}$$

The administrator can identify $w^* = w_i$ by testing if the δ_i is an n-th power, which can be realized by zero knowledge proof as follows:

1) The investigator chooses a random number $\rho \in Z^*_n$, computes $a_\rho = \rho^n \ mod\ n^2$ and sends a_ρ to the administrator;
2) The administrator chooses a random bit string $\mathcal{S}$ of length e and sends $\mathcal{S}$ to the investigator, where $\mathcal{S} < 2^e$ and $2^e < \min(p, q)$;
3) The investigator computes $\mu = \rho \cdot r^{\mathcal{S}} \ mod\ n$ and sends μ to the administrator;
4) The administrator verifies if $\mu^n = a_\rho \cdot \delta_i^{\mathcal{S}} \ mod\ n^2$, where $\mu^n = (\rho \cdot r^{\mathcal{S}})^n = \rho^n \cdot r^{n\mathcal{S}} \ mod\ n^2$ only if $\delta_i = r^n \ mod\ n^2$ is true, in other words, $\mu^n = a_\rho \cdot \delta_i^{\mathcal{S}} \ mod\ n^2$ holds only if $w^* = w_i$.

After identifying $w^* = w_i$ both of which are in an encrypted form, the administrator returns the investigator $E(W)$ which contains $E(w^*)$. The above is constant-round zero knowledge

proof, so the investigator and administrator can prepare the parameters (such as a_ρ, $\mathcal{S}$ and μ) in advance for the sake of efficiency. In addition, the purpose of the search is to find documents which contain a specific word, where the position and the number of occurrences are not relevant in our proposal. So, as long as that δ_i is an n-th power is identified, the server administrator will stop searching on this document and continue to search on next document.

IV. A SCHEME BASED ON COMMUTATIVE ENCRYPTION

In the above scheme, we must trust the server administrator to return all the searching results. If the server administrator returns only some (but not all) of the searching results, the investigator will have no way to detect this. Until now, we assume that the administrator does not misbehave in this way and trustable in this sense.

We can eliminate such assumption by introducing a Trusted Third Party (TTP), which is in charge of searching and supervising the server administrator returning all the search results. Suppose that p_I and p_A are public keys of the investigator and the server administrator, respectively. We write $E_k(m)$ for the result of encrypting m with k and define that E_{p_I} and E_{p_A} are commutative encryption if

$$E_{p_A}(E_{p_I}(m)) = E_{p_I}(E_{p_A}(m)) \tag{5}$$

holds for any plaintext m. Our second scheme is based on such commutative encryption and the detail is the following.

1) To keep the investigation subject secret from the TTP, the investigator uses p_I to encrypt the keyword w^* and provides the TTP the encrypted keyword $E_{p_I}(w^*)$;
2) To keep the server data secret from the TTP, the server administrator uses p_A to encrypt the document W and provides TTP the encrypted document $Ep_A(W)$ which has the following form:

$\cdots$	$Ep_A(w_{i-1})$	$Ep_A(w_i)$	$Ep_A(w_{i+1})$	$\cdots$

3) The TTP encrypts $E_{p_I}(w^*)$ with p_A to obtain $E_{p_A}(E_{p_I}(w^*))$ and encrypts $E_{p_A}(w_i)$ with p_I to obtain $E_{p_I}(E_{p_A}(w_i))$. As E_{p_I} and E_{p_A} are commutative encryption, $E_{p_A}(E_{p_I}(w^*))$ and $E_{p_I}(E_{p_A}(w_i))$ will be equal if w^*=w_i; without knowing the values of w^* and w_i, the TTP compares $E_{p_A}(E_{p_I}(w^*))$ with $E_{p_I}(E_{p_A}(w_i))$; As the two values are equal, the TTP will supervise the administrator decrypting the document $E_{p_I}(E_{p_A}(W))$ (=$E_{p_A}(E_{p_I}(W))$ which contains the specified keyword w^* and returning the relevant $E_{p_I}(W)$ to the investigator.
4) The investigator decrypts $E_{p_I}(W)$ with his private key and investigate W for capturing the evidence. That is, the investigator performs investigation only on the relevant data.

We construct commutative encryption based on matrix polynomial (i.e., a polynomial with matrices as variables) in $Z_q[x]$, where q is a large prime. For example, let $h(x)$=$a_0 + a_1x + \ldots + a_nx^n \in Z_q^n[x]$ is a polynomial in x, where a_0,a_1,...,a_n are constants. We can easily obtain a matrix polynomial $h(A)$=$a_0I + a_1A + \ldots + a_nA^n \ mod\ q$, where A is a square matrix and I is the identity matrix with the same size of A. Obviously, the multiplication of any two matrix polynomials is commutative.

We take the investigator as an example to demonstrate the procedures of encryption and decryption. The investigator can select a polynomial $f(x)\in Z_q^n[x]$ and a square matrix R to compute $f(R)$, where $f(R)$ is invertible and its inverse is denoted by $f(R)^{-1}$. Not all the matrices have an inverse and the inverse of $f(R)$ is very hard to evaluate as the size of matrix is large, that is where some elementary linear algebra comes in. So the investigator can take $f(R)$ and $f(R)^{-1}$ as his public key p_I and private key s_I respectively. Rewrite the plaintext m in a vector or matrix form and the investigator performs the encryption by computing

$$c = E_{p_I}(m) = E_{f(R)}(m) = f(R)m \ \ mod\ q \tag{6}$$

and performs the decryption by computing

$$m = E_{s_I}(c) = E_{f(R)^{-1}}(c) = f(R)^{-1}c \ \ mod\ q \tag{7}$$

Similarly, the administrator can also select a polynomial $g(x)\in Z_q^n[x]$ and a square matrix T (with same size of R) to make $g(T)$ invertible, and then perform the encryption and decryption. It is obvious that $f(R)g(T)$=$g(T)f(R)$ holds, which means such encryption is commutative.

V. DISCUSSION AND CONCLUSION

A. Discussion

To improve the efficiency in forensic investigation, the investigator is supposed to capture evidence only from the relevant data in our proposals. Through searching for encrypted keywords (specified by investigation subject) on encrypted data (stored on the server), we realized that the investigator can search for evidence without learning any information of irrelevant data and the server administrator cannot learn the investigation subject. Obviously, whether the privacy on both sides can be completely protected relies on the security of cryptosystem.

In the above two schemes, we assumed that the file can be easily broken into a sequence of words of a fixed length. However, this assumption might not be true in a normal file. To deal with variable-length words, we can pick a fixed-size block that is long enough to contain most words like the work [1], 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.

B. Conclusions

Based on homomorphic encryption and commutative encryption, we presented two schemes to assist investigators in searching for evidence efficiently without exposing innocent data to the investigators. For future work, we will consider how to implement them and verify their feasibility.

REFERENCES

[1] D. Song, D. Wagner, and A. Perrig, *Practical Techniques for Searches on Encrypted Data*, in Proceedings of IEEE Symposium on Security and Privacy 2000, pp.44-55, 2000.

[2] Y.C. Chang, and M. Mitzenmacher, *Privacy Preserving Keyword Searches on Remote Encrypted Data*, Cryptology ePrint Archive, Report 2004/051, Feb 2004.

[3] Reza Curtmola, Juan Garay, Seny Kamara and Rafail Ostrovsky, *Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions*, in CCS, pp.79-88, 2006.

[4] Eu-jin Goh, *Secure Indexes*, in the Cryptology ePrint Archive, Report 2003/216, March 2003.

[5] P. Golle, J. Staddon, and B. R. Waters, *Secure Conjunctive Keyword Search over Encrypted Data*, in Proc. of ACNS'04, pp.31-45, 2004.

[6] Hasan Al-Sakran and Mohamad Adnan Ali, *Designing Efficient Techniques for Searching Encrypted Data in Untrusted Infrastructure*, Information Technology Journal 5(2), pp.347-352, 2006.

[7] Lee Thian Aun Joseph, Azman Samsudin, and Bahari Belaton, *Efficient Search on Encrypted Data*, 13th IEEE International Conference on Networks, pp.352-357, 2005

[8] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano, *Public Key Encryption with Keyword Search*, in Proceedings of Eurocrypt 2004, Lecture Notes in Computer Science 3027, pp.506-522, 2004.

[9] Dan Boneh, Eyal Kushilvitz, Rafail Ostrovsky, and William E. Skeith III, *Public Key Encryption That Allows PIR Queries*, CRYPTO 2007, pp.50-67, 2007.

[10] G.Sammour, M.Shareef, and K.Kaabneh, *Heuristic Search on Encrypted Data (HSED)*, EBEL, 2005.

[11] Maisa Halloush, and Mai Sharif, *Global Heuristic Search on Encrypted Data (GHSED)*, IJCSI International Journal of Computer Science Issues, Vol. 2, pp.13-17, 2009.

[12] B.Chor, O.Goldreich, E.Kushilevitz, and M.Sudan, *Private Information Retrieval*, in Proc. of 36th FOCS, pp.41-50, 1995.

[13] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin, *Protecting Data Privacy in Information Retrieval Schemes*, JCSS, pp.151-160, 1998.

[14] R.Ostrovsky, and W.Skeith, *Private Searching on Streaming Data*, Lecture Notes in Computer Science, vol.3621, pp.223-240, 2005.

[15] Caroline Fontaine, and Fabien Galand, *A Survey of Homomorphic Encryption for Nonspecialists*, EURASIP Journal on Information Security, vol.2007, pp.1-10, 2007.

[16] Pascal Paillier, *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, Advances in Cryptology EUROCRYPT 99, vol.1592, pp.223-238, 1999.