Secure Search Over Encrypted Data in Cloud Computing

Majid Bakhtiari
Department of
Computer Science
Faculty of Computing
Universiti Teknologi Malaysia
UTM Skudi, 81310
Johor, Malaysia
e-mail: bakhtiari@utm.my

Majid Nateghizad
Department of
Computer Science
Faculty of Computing
Universiti Teknologi Malaysia
UTM Skudi, 81310
Johor, Malaysia
e-mail: nmajid2@live.utm.my

Anazida Zainal
Department of
Computer Science
Faculty of Computing
Universiti Teknologi Malaysia
UTM Skudi, 81310
Johor, Malaysia
e-mail: anazida@utm.my

Abstract*—Nowadays cloud computing is widely used by users and organization to get benefit of many services in cloud. Huge datasets in cloud persuade users to outsource their information and documents to the untrusted Cloud Service Provider (CSP). On the other hand, there are some privacy and security problems in cloud storage considering as main drawbacks of extending it among users. One of the solutions for providing confidentiality of data in cloud storage is encrypting data before sending to cloud. This satisfies data confidentiality and makes users feel more confidence on the CSP. However, this prevents user to search the outsourced documents directly. Regular encryption algorithms such as AES, RC4 and DES mechanisms have searching limitation; in which the whole ciphertext needs to be retrieved and then decrypt before search procedure can be performed. Recently a lot of research has been done to enable search capabilities for users. Generally, keyword based search approach is used. This approach allows users to retrieve just those documents contain special keywords. However, searchable encryption algorithms suffer from privacy and security problems. This paper proposes Secure Searchable Based Asymmetric Encryption (SSAE) algorithm that provides Indistinguishability under Adaptive-Chosen Ciphertext Attack. The proposed searchable encryption algorithm is mathematically proven secure and it has the ability to perform a search within encrypted data without decrypting them.

Keywords-component; cloud computing; searchable encryption; keyword; security; asymmetric

I. INTRODUCTION

Nowadays people have a lot of information to store in electronic devices. Cloud storage is the new technology that provides many services such as remote storage, which makes users able to access their information everywhere through the Internet in a cost effective manner regardless of servers' location. Privacy and security issues are among the most critical problems of the cloud storage. These problems are considering drawbacks in extension of the cloud storage around the globe[1-6]. For the sake of providing more security and privacy of outsourced documents, encrypting data before outsourcing consider as the best solution that makes cloud storage more reliable for users[7]. This approach can be applied by any encryption algorithms such as AES, RC4 and DES[8, 9]. Despite many advantages of encryption, this makes new challenges for users to manage their own outsourced data. For example user may want to retrieve only information contains set of specific keywords.

Searchable encryption is a cryptography primitive that enables users to search through outsourced encrypted data without exposing keywords to the untrusted server[10-12]. The area of searchable encryption has been known by The Defense Advanced Research Projects Agency(DARPA) as one of the technical advances for balancing between security and privacy. Every searchable encryption must have three characteristics[13-15]:

- Encryption algorithm must provide a high level of security against different types of attacks to make sure there is no data exposure.
- Searchable encryption algorithms should have the ability to search an encrypted keyword in outsourced ciphertext on CSP directly (without decrypting documents) that reveal nothing confidential.
- Search procedure of searchable encryption algorithm should provide data confidentiality meaning CSP should not learn about contents of the encrypted data during the search process; for example, two documents contain the same encrypted keyword after user performs several queries. Therefore, not only the keyword should be encrypted during the search, also method of the search should be secure.

There are four categories of ciphertext indistinguishability[16-18]:

- Indistinguishability under Chosen Plaintext Attack(IND-CPA)
- Indistinguishability under (Non-Adaptive) Chosen Ciphertext Attack(IND-CCA)

978-1-4799-2758-6/13 $31.00 © 2013 IEEE
DOI 10.1109/ACSAT.2013.64

- Indistinguishability under Adaptive Chosen Ciphertext Attack(IND-CCA2)

A. *Indistinguishability under chosen plaintext attack*

IND-CPA security intuitively captured the notion that an attacker has the ability to see specific messages sent between the two communicating parties. For a probabilistic asymmetric key encryption algorithm, the following scenario between an adversary and a challenger defines indistinguishability under chosen plaintext attack. In computational security based schemas, a probabilistic polynomial time Turing machine models the adversary, meaning that it must output a "guess" and complete the game within a polynomial number of time steps. In this definition, E (P_K, M) shows the encryption of a message M under the key P_K.

Based on security parameter k (e.g., a key size in bits), the challenger generates a key pair P_K and S_K, and sends P_K to the adversary but he retains S_K. Performing any number of encryptions or other operations from the adversary part is allowed. Eventually, the adversary submits two distinct chosen plaintexts M_0 and M_1 to the challenger. A selected bit b ϵ {0, 1} uniformly at random from challenger as a challenge ciphertext C = E (P_K, M_b) goes back to the adversary. The adversary can perform any number of additional computations or encryptions. Finally, it outputs a guess for the value of b.

Indistinguishability under chosen plaintext attack is also secure against Known Plaintext Attack(KPA).In KPA the attacker has samples of both the plaintext and its encrypted version (ciphertext). These can be used to reveal further secret information such as secret keys and codebooks.

B. *Indistinguishability under chosen ciphertext attack*

The IND-CCA security game is the same as the IND-CPA game, except that now the attacker has access to a decryption oracle in addition to the encryption oracle. Note that IND-CCA security includes IND-CPA security because an attacker can always choose not to use the decryption oracle. In asymmetric key encryption algorithm, indistinguishability under chosen ciphertext attack is defined by game between an adversary and a challenger. The procedural steps for the game are listed below:

1. Challenger runs KeyGen

2. Attacker is given two oracles, Enc (K,.) and Dec (K, .) (Query Phase I)

3. Attacker produces two messages M_0 and M_1. The challenger returns the challenge ciphertext C^*=Enc(K, M_b)(Challenge phase)

4. Same as Query Phase I except cannot query the decryption oracle on C^* (Query Phase I)

5. Attacker outputs b′

The advantage of attacker A in IND-CCA game is Adv_A=Pr[b= b′] - 1/2.

C. *Indistinguishability under adaptive chosen ciphertext attack (IND-CCA2)*

Another attack scenario defined under IND-CCA2 term is that the adversary tries to gain partial information by making queries to the decryption oracle based on the results of previous decryption. Indistinguishability under non-adaptive and adaptive Chosen Ciphertext Attack uses a definition similar to that of IND-CPA. However, in addition to the public key, the adversary is given access to a decryption oracle, which decrypts arbitrary ciphertexts at the adversary's request, returning the plaintext.

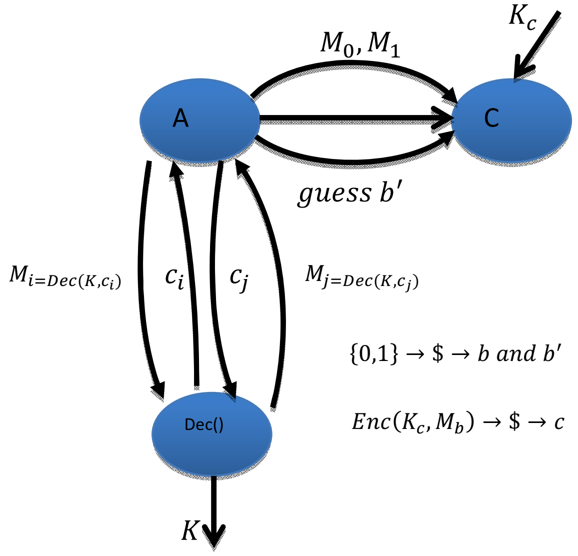

Figure 1. IND-CCA and IND-CCA2

In the non-adaptive definition, the adversary is allowed to query this oracle only until it receives the challenge ciphertext. In the adaptive definition, the adversary may continue to query the decryption oracle even after it has received a challenge ciphertext, with the caveat that it may not pass the challenge ciphertext for decryption.

Fig. 1 shows IND-CCA and IND-CCA2 process where adversary A attacks challenger C. In the IND-CCA and IND-CCA2 security approaches, adversary A may perform any number of encryptions and calls the decryption oracle Dec(). Adversary in these two security approaches can perform any number of additional encryptions. The difference between IND-CCA and IND-CCA2 in fig. 1 is that in IND-CCA, adversary cannot make further calls to decryption oracle but in IND-CCA2, adversary can make further calls to decryption oracle but may not submit challenge ciphertext c.

In the IND-CCA2 security game, challenger generates a key pair P_K, S_K based on some security parameter K (e.g., a key size in bits), and publishes P_K to the adversary. The challenger retains S_K. The adversary may perform any number of encryptions, calls to the decryption oracle based on arbitrary ciphertexts, or other operations. Eventually, the adversary submits two distinct chosen plaintextsM_0, $M_1$to the challenger. The challenger selects a bit b ϵ {0, 1} uniformly at random and sends the "challenge" ciphertext c = E (P_K, M_b) back to the adversary. The adversary is free to perform any number of additional computations or encryptions. In the non-adaptive case (IND-CCA), the adversary may not make further calls to the decryption oracle. Meanwhile in the adaptive case (IND-CCA2), the adversary may make further calls to the decryption oracle, but may not submit the challenge ciphertext c. Finally, the adversary outputs a guess for the value of b .

Table 1. Ciphertext Indistinguishability

Attack/Security Mechanism	IND-CPA	IND-CCA	IND-CCA2
KPA	✓	✓	✓
CPA	✓	✓	✓
CCA	✗	✓	✓
CCA2	✗	✗	✓

Table 1 illustrates the different among different security approaches. According to the Table 1, the IND-CPA security approach is unsecure known non-adaptive chosen ciphertext attack and adaptive chosen ciphertext attack. In the other hand, IND-CCA2 is resistant to known plaintext attack, chosen plaintext attack, non-adaptive chosen ciphertext attack, and adaptive chosen ciphertext attack.

In this paper, we propose a new searchable encryption scheme, which addresses some of the above problems. The contribution of our scheme can be summarized as follows:

i. The proposed searchable encryption supports keyword searching on encrypted data. It enables users to search desired word through ciphertext on CSP without revealing any information to the CSP.
ii. The proposed encryption scheme is semantically resistant against adaptive chosen ciphertext attack(IND-CCA2). In addition, it is proved under the Bilinear Diffie-Hellman to b4e secured[16].

This paper is organized as follows: It begins by reviewing some related works and explains preliminaries. Later, it presents an outline of the proposed encryption scheme. Finally, this paper discusses the construction of the proposed encryption scheme and its security.

II. PEKS ALGORITHM

The proposed secure searchable encryption algorithm is this paper is based on Public Key Encryption With Keyword Search(PEKS)[19]. For the sake of better understanding the proposed algorithm in this paper, PEKS algorithm and structure are discussed in this section.

a) KeyGen will generate a public key and private key for Alice. A user A uses the public key in order to encrypt tokenized words W′ in searchable way. To perform a search for desired word A uses his private key.
b) The PEKS function encrypts word of documents with user's public key.
c) When user A wants to retrieve documents that contain specific word W, he performs search algorithm, which makes a trapdoor of W using private key to make it searchable within encrypted outsourced data in CSP.
d) After making trapdoor of W in Alice's computer, A sends it to CSP to perform search.
e) Email server compares trapdoor given from A and encrypted keywords W′ to find document containing W.

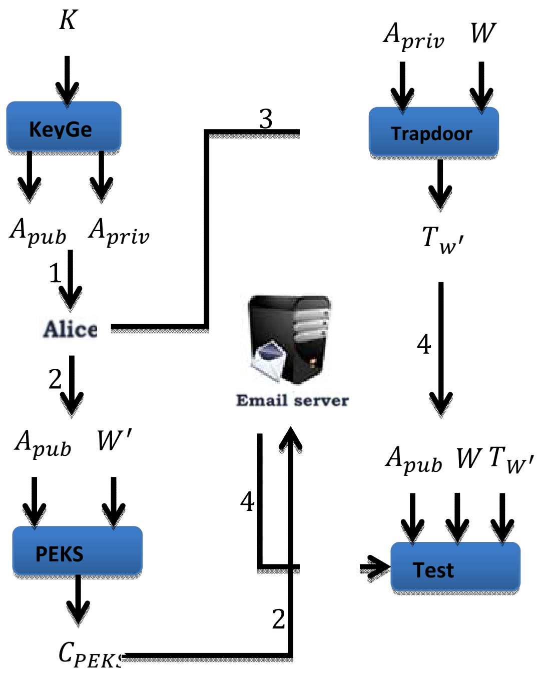

Figure 2. PEKS Algorithm

Fig.2 shows PEKS algorithm steps. These steps are explained more in details as follows:

A. KeyGen (K):

Takes a security parameter *K* as input, *and generates a* public key and private key for (A_{pub} ,A_{priv}) for Alice.

B. PEKS (A_{pub}, W′):

In this function, PEKS encrypts word W′ with given public key A_{pub} and outputs ciphertext of W′ C_{PEKS}. The encryption algorithm should apply on W′ in searchable mode.

C. Trapdoor (A_{priv}, W):

For searching through encrypted words, this function produces trapdoor of word W which user wants to search. This function works with given private key A_{priv} and outputs T_W.

D. Test (A_{pub}, S, T_W):

Given Alice's public key $A_{pub,}$ C_{PEKS} = PEKS (A_{pub}, W′) and a trapdoor T_W= Trapdoor(A_{priv}, W), if W=W′ the output is equal to 'Yes'. Otherwise it outputs 'No'.

III. Outline of the proposed searchable encryption algorithm

In this section, the outline of proposed secure searchable encryption algorithm is described. However, before explaining the outline of the algorithm, the assumptions are needed to be stated. In this section, the outline of proposed secure searchable.

A. Assumptions

1. Assume *G* and *G1* is a set of prime order, *p* in which there are computable bilinear map from *G* into *G1*.
   a) Computable: *e* (*g*, *h*) ϵ*G1* for given *g*, *h*ϵ*G* takes a polynomial time.

   b) Bilinear: any integers x , *y*ϵ [1,*p*] should satisfy e(g^x,g^y)=e$(g,g)^{xy}$

   c) Non-degenerate: e (g, g) ≠1 and also*e* (*g*, *g*) must be a generator of *G*1 if g be a random generator of *G*.
2. Decisional BDH Assumption: Assume that challenger chooses four random values a, b, c and d ϵZ_p and other random binary value ß. Hence, there are two possibilities of ß, if ß=1 then it outputs are (g, A=g^a, B=g^b, C=g^c, z=$e(g,g)^{abc}$), otherwise the outputs are (g, A=g^a, B=g^b, C=g^c, z=$e(g,g)^z$). In this scenario an adversary has at least ε advantage to solve BDH problem where |PR[ß(g, g^a, g^b, g^c, $e(g,g)^{abc}$) = 1] - PR[ß(g, g^a, g^b,g^c,$e(g,g)^z$) = 1] | ≥ 2ε.

Computational BDH Assumption: randomly chosen a, b ϵ Z_p outputs (g, A =g^a, B=g^b).It is about advantages of adversary A in attempting to output [g^{ab}ϵ G is ε if PR [ß(g, g^a, g^b) = g^{ab}] ≥ ε

B. Outline of proposed algorithm

Assume that a user needs to outsource critical data in CSP. Those data contains a series of messages m ϵ M. In order to make this encryption searchable, we need to send encrypted keywords to the CSP along with encrypted data.

For encrypting data and keywords, our proposed algorithm uses public key A_{pub} and for decrypting information or generating trapdoor, it uses two different private keys.

a) *KeyGen(s):* Given security parameter *s* and computes A_{pub} and A_{priv} as public/private keys.

b) *Encryption(A_{pub}, m):* Given public key A_{pub} and message M, encrypt the message M using A_{pub}

c) *SEncryption (A_{pub}, W):* Encrypt given word W in searchable form by using public key A_{pub}.

d) *Trapdoor (A_{priv}, W):* Takes private key A_{priv} and word W to generate trapdoor T_W.

e) *Test (A_{pub}, S, T_W):* Search within searchable encrypted data S=PEKS (A_{pub}, W') in CSP for a given trapdoor T_W and word W. It outputs yes if W= W'meaning this message containing word W.

f) *Decryption (A_{priv}, E):* it takes private key A_{priv} and encrypted message E to recover plain text m ϵ M.

IV. CONSTRUCTION OF THE PROPOSED SEARCHABLE ENCRYPTION ALGORITHM

The proposed schema is based on bilinear map. Assume that we have a function that generates the required parameters G, G1, α, g, g_2 and u' where G and G1 are two groups of prime order *p* that a bilinear map e: G*G→G1 must satisfied. Meanwhile *h* is another computed variable based on *g* in which it takes a polynomial time to compute e(*g*,*h*) ϵ G1, e(g^x,g^y) = $e(g,g)^{xy}$ and *g* is a random generator of G which e(g,g) is a generator of G1. In the proposed algorithm two hash functions are used H_1: {0,1}* → G and H_2: G1→ $\{0,1\}^{(\log p)}$.

A. KeyGen:

For $i = 1, \dots, d$ run algorithm G(s) (here G is a key generation algorithm) to generate a public and private key pair $PK_i/Priv_i$for the source-indistinguishable encryption schema. Then public key is $A_{pub} = \{PK_1, \dots, PK_d\}$. The private key is $A_{priv} = \{Priv_i, \dots, Priv_d\}$. A q-uniform t-cover free family of substance $F = \{S_1, \dots, S_k\}$of $\{PK_1, \dots, PK_d\}$. Hence, each S_i is a subset of public keys.the proposed algorithm is identity based, *v* is an *n* bits that stands for an identity and V_is the i^{th} bit of *v* where *i*⊆ {1,…, n} in which v_i= 1. Finally, KeyGen outputs a private key to identify *v* as follows: d_1=$g_2^\alpha(u' \prod_{i\in v} u_i)^r$ and d_2= g^r, $d_v = (d_1, d_2)$or $d_v = (g_2^\alpha(u' \prod_{i\in v} u_i)^r, g^r)$.

B. Encryption:

In this step the message M ϵ G1 is encrypted for an identity by assuming that t ϵZ_p is a random number. The generated encrypted message is as follows:

There are three parametersC_1, C_2, C_3 that will be sent to CSP. $C_1 = e(g_1, g_2)^t M$, $C_2 = g^t$, $C_3 = (u' \prod_{i\in v} u_i)^t$ and finally $C = (C_1, C_2, C_3)$ or $C = e(g_1, g_2)^t M, g^t, (u' \prod_{i\in v} u_i)^t)$.

C. SEncryption (A_{pub}, W):

Let $S_i \in F$ be the subset associated with the word $W_i \in \sum$. Let $S_i = \{PK^{(1)}, \dots, PK^{(q)}\}$. Pick random message $M_1, \dots, M_q \in \{0,1\}^s$ and let $M = M_1 \oplus \dots \oplus M_q$ and. It outputs $SEncryption(A_{pub}, W_i) =$ $(M, E[PK^{(1)}, M_1], \dots, E[PK^{(q)}, M_q])$.

D. Trapdoor (A_{priv}, W):

Let $S_i \in F$ be the subset associated with the word $W_i \in \sum$. The trapdoor for word W_i is simply the set of private keys that correspond to the public keys in the set S_i.

E. Test (A_{pub}, S, T_W):

Let $T_W = \{Priv^{(1)}, \dots, Priv^{(q)}\}$ and let $R = (M, C_1, \dots, C_q)$ be a $SEncryption$. For $i = 1, \dots, q$ decrypt each C_i using private key $Priv^{(i)}$ to obtain M_i. Output 'Yes' if $M = M_1 \oplus \dots \oplus M_q$, and output 'no' otherwise.

F. Decryption:

In the encryption process, three parameters are sent to CSP. They are $e(g_1, g_2)^t M, g^t$ and$(u' \prod_{i \in v} u_i)^t$, assume C_1=$e(g_1, g_2)^t M$, C_2=g^t and C_3=$(u' \prod_{i \in v} u_i)^t$

$$M = \ C_1 \frac{e(d_2, C_3)}{e(d_1, C_2)} =$$

$$e(g_1, g_2)^t M \left(\frac{e\left(g^r, (u' \prod_{i \in v} u_i)^t\right)}{e(g_2^{\alpha}, (u' \prod_{i \in v} u_i)^r, g^t)} \right) \text{ Where ;}$$

$$\frac{e(g^r, (u' \prod_{i \in v} u_i)^t)}{e(g_2^{\alpha}, (u' \prod_{i \in v} u_i)^r, g^t)} = \frac{e\left(g, (u' \prod_{i \in v} u_i)^{rt}\right)}{e(g_1, g_2)^t e((u' \prod_{i \in v} u_i)^{rt}, g)}$$

$$= M$$

V. SECURITY OF THE PROPOSED SEARCHABLE ENCRYPTION ALGORITHM

To evaluate the security of proposed algorithm this paper analyzes two different security approaches identity-based encryption and bilinear Diffie-Hellman problem.

A. Identity-Based Encryption

This definition was first described by Boneh and Franklin[19]. Consider the following game played by an adversary. The game has five steps:

Step1: The client generates the master public parameters (public key and parameter) and then it will broadcast in the network. In this case, attacker can capture the master public parameters.

Step2: The attacker is allowed to make a query for a private key, where private key is an identity specified by the attacker. The attacker can repeat this procedure numerous times for different identities.

Step3: The attacker submits a public key, and two messages $M_0$and M_1.The attacker's choice of public key is restricted to the identities that he did not request a private key in step 2. The challenger flips a fair binary coin, and returns an encryption of Mγ under the public key.

Step4: In this step, step 2 is repeated with the restriction that the attacker cannot request the private key for the public key.

Step5: The attacker submits a guess,γ_0, of γ .

An Identity Based Encryption scheme is (t, q, ε) semantically secure assuming (t + o(ε^{-2} ln(ε^{-1})λ^{-1} ln(λ^{-1})), ε/32(n+1)q) where λ=1/8(n+1)q if all t times adversaries making at most q private key queries have at most an ε in breaking IBE scheme.

B. Bilinear Diffie-Hellman Problem

Let g be a random generator of G, the problem of bilinear Diffie-Hellman is defined as follows: given g, g^a,g^b, $g^c \epsilon$G as input of the scenario and also compute $e\ (g, g)\ ^{abc} \epsilon$ G2.In solving BDH, if all polynomial time algorithms have a negligible advantage, BDH is intractable. It is also proved that the non-interactive searchable encryption algorithm PEKS against chosen keyword attack is secure, assuming BDH is intractable[20].

VI. CONCLUSION

Nowadays CSPs offer many services in cloud that can benefit people and organizations to reduce their Business management cost. However, using cloud storage brings some privacy issues[1-3]. This paper highlighted those issues of the outsourcing data in cloud storage.This paper proposed a new practical searchable encryption algorithm, which has proven to improve the security of searchable encryption from IND-CPA to IND-CCA2.However, this algorithm suffers from high computational cost in searching process. A lot of research has been done topropose an index based search algorithm within encrypted documents.

It is possible to merge the Waters IBE[21]algorithm with the index based searchable encryption proposed by Curtmula[15] to get better performance in searching through outsourced encrypted documents. The construction of the algorithm is based on combination of look-up table T and array A that can be as follows: first, all the unique words in each document are tokenized in an array. Then we create a link list that contains a list of documents, which contains the same keyword. Once we make link lists of all unique keywords for all documents, they will be compressed, encrypted and mixed up into array A. Since elements are encrypted with different keys, the key of the next element is located in each element. The algorithm generates a look-up table, which contains all the encrypted unique keywords and the addresses of documents containing those keywords in array A. Even though this algorithm performs searching in constant time that can be a major advantage in large dataset, there is a major problem in the algorithm. It needs to update array A, trapdoor T_W and construct look-up table T whenever a document is added or removed from database. Another drawback is that a unique array is used for all documents, which increases the time for updating a single element.

Our future work is to investigate other search methods and propose a secure practical index based search algorithm over encrypted data that satisfies a reasonable performance concerning cost of search.

REFERENCES

[1] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*: O'Reilly, 2009.

[2] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 2010, pp. 693-702.

[3] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications,* vol. 34, pp. 1-11, 2011.

[4] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," *Security & Privacy, IEEE,* vol. 8, pp. 24-31, 2010.

[5] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," in *Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on*, 2010, pp. 105-112.

[6] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems,* vol. 28, pp. 583-592, 2012.

[7] S. Zhang, S. Zhang, X. Chen, and X. Huo, "Cloud computing research and development trend," in *Future Networks, 2010. ICFN'10. Second International Conference on*, 2010, pp. 93-97.

[8] N. Standard, "Data Encryption Standard," *Federal Information Processing Standards Publication,* 1999.

[9] A. Mousa and A. Hamad, "Evaluation of the RC4 Algorithm for Data Encryption," *IJCSA,* vol. 3, pp. 44-56, 2006.

[10] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 451-459.

[11] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 2011, pp. 383-392.

[12] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, 2010, pp. 253-262.

[13] P. Van Liesdonk, S. Sedghi, J. Doumen, P. Hartel, and W. Jonker, "Computationally efficient searchable symmetric encryption," in *Secure Data Management*, ed: Springer, 2010, pp. 87-100.

[14] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Advances in Cryptology-CRYPTO 2007*, ed: Springer, 2007, pp. 535-552.

[15] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 79-88.

[16] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO 2001*, 2001, pp. 213-229.

[17] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Advances in Cryptology—CRYPTO'99*, 1999, pp. 537-554.

[18] E. Kiltz and J. Malone-Lee, "A general construction of IND-CCA2 secure public key encryption," in *Cryptography and Coding*, ed: Springer, 2003, pp. 152-166.

[19] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-Eurocrypt 2004*, 2004, pp. 506-522.

[20] D. Boneh, "The decision diffie-hellman problem," in *Algorithmic number theory*, ed: Springer, 1998, pp. 48-63.

[21] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology–EUROCRYPT 2005*, ed: Springer, 2005, pp. 114-127.