International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

Ensures Data Security and Message Integrity in Cloud using Hibernation

L. Muralidaran[1] T. Vidhya[2] K. Swathy[3] K. Kanimozhi[4]

Assistant Professor, Dept. of CSE, Christ college of Engineering & technology, Puducherry.
U.G Student, Dept. of CSE, Christ college of Engineering & technology, Puducherry.
U.G Student, Dept. of CSE, Christ college of Engineering & technology, Puducherry.
U.G Student, Dept. of CSE, Christ college of Engineering & technology, Puducherry

Abstract

Cloud computing where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, so that users can resort to an external audit party to check the integrity of outsourced data when needed. Here we present a technique for distributing trust-needing computation onto insecure networks, while providing probabilistic guarantees that malicious agents that compromise parts of the network cannot learn private data. As security is concerned, We implies Elliptic Curve Cryptography (ECC) algorithm along with hashing technique for ensuring security for the message to be sent along with acknowledge.

Keywords: **Cloud computing, data privacy, ECC, hibernation, hashing.**

1. Introduction

Cloud Computing has been envisioned as the next-generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced into the Cloud. While these advantages of using clouds are unarguable, due to the opaqueness of the Cloud—as separate administrative entities, the internal operation details of cloud service providers (CSP) may not be known by cloud users—data outsourcing is also relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons.

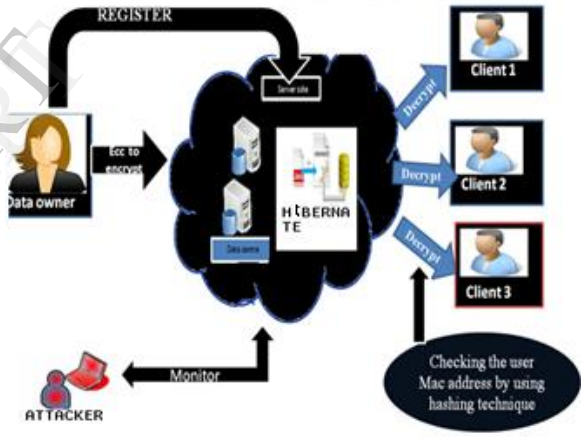

Fig 1: Network Architecture for the Cloud Users

We provide secure and privacy-preserving access control to users, which guarantees any member in a group to anonymously utilize the cloud resource. And also we provide rigorous security analysis, and perform extensive simulations to demonstrate the efficiency of our scheme in terms of storage and computation overhead. The security and confidentiality for our document is provided by ECC algorithm. The discrete problem on elliptic curve groups is believed to be more difficult to trace when compared to other algorithms. Using the finite fields we can form an Elliptic Curve Group where we also have a DLP problem which is harder to solve so that

the intruder cannot find the traces easily, which provides a high level of security.

1. "Elliptic" is not elliptic in the sense of a "oval circle".
2. "Curve" is also quite misleading if we're operating in the field F_p. The drawing that many pages show of a elliptic curve in R is not really what you need to think of when transforming that curve into F_p. Rather than a real "curve" (i.e. a not-straight line), it is more like a cloud of points in the field -- these points are not connected. The ECC equation is

$$y^2 = x^3 + ax + b \mod p$$

Plotting multiple things is done by adding plots in Sage. Consequently we will be able to use the sum-function to plot all points

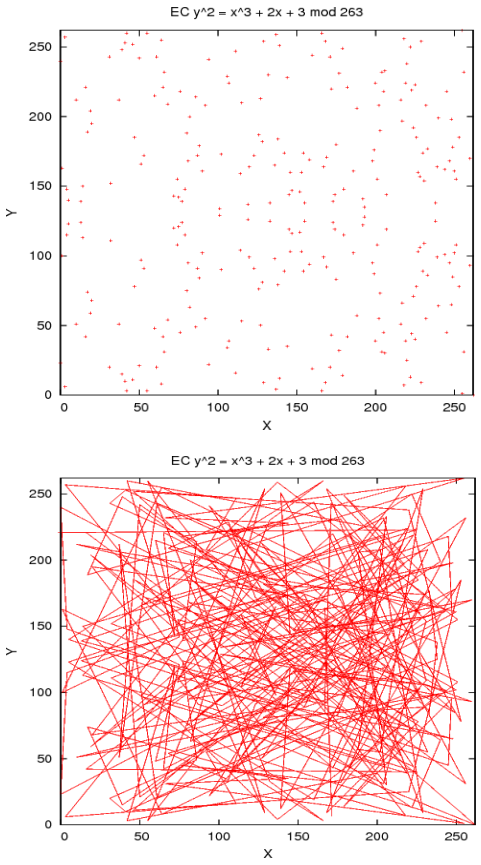

Fig 2: Plotted points along with subgroups

We provide secure and privacy-preserving access control to users, which guarantees any member in a group to anonymously utilize the cloud resource. And also we provide rigorous security analysis, and perform extensive simulations to demonstrate the efficiency of our scheme in terms of storage and computation overhead.

- In precise, our proposed system includes the concept of Elliptical Curve Cryptography Algorith(ECC) to provide the high level of security by using graphical plots or curves. This is done since discrete logarithm problem on elliptic curve groups is believed to be more difficult than the corresponding problem in (the multiplicative group of nonzero elements of) the underlying finite field.

Then it is followed by hashing concept which is used here as a tool to identify the intruders trying to decrypt our entire system. Then comes hibernate which maintains our database persistent and reduces the number of hits.

1.1 ECC Algorithm

Alice

Private key *dA*, Public key *QA=dAP*.

Signature generation

1.Select a random *k* from [1, *n*-1]
2.Compute *kP*=(*x*1,*y*1) and *r=x1mod n*. if *r*=0 goto step 1
3.Compute *e=H*(*m*), where *H* is a hash functon, *m* is the message.
4.Compute *s=k-1*(*e+dAr*) *mod n*. If *s*=0 go to step 1.
(*r, s*) is Alice's signature of message *m*

Bob

Signature verification

1.Verify that *r, s* are in the interval [1,*n*-1]
2.Compute *e=H*(*m*), where *H* is a hash functon, *m* is the message.
3.Compute *w=s-1 mod n*
4.Compute *u1=ew mod n* and *u2=rw mod n.*
5.Compute *X=u1P+u2QA*=(*x*1,*y*1)
6.Compute *v=x1 mod n*
7.Accept the signature if and only if
v=r
m, r ,s

- An elliptic curve over a field ***K*** is a nonsingular cubic curve in two variables, *f(x,y) =0* with a rational point (which may be a point at infinity).

- Elliptic curves groups for cryptography are examined with the underlying fields of F_p (*where p>3 is a prime*) and F_2^m (*a binary representation with* 2^m *elements*).
- Any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems

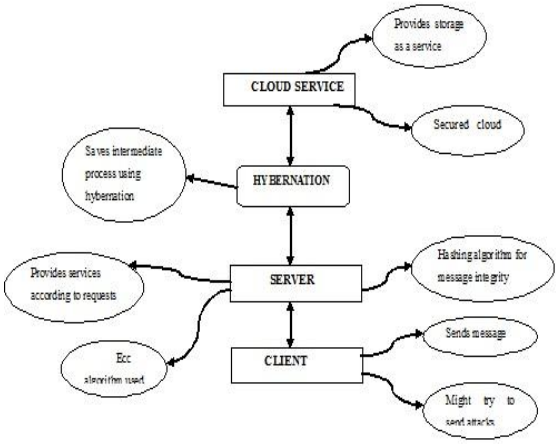

Fig 3: Entire proposed system architecture

2. Related Work

In [1] compared the different schemes in the Cloud computing field. However, concerns of sensitive information on cloud potentially causes privacy problems. In this model, Data encryption protects data security to some extent, but at the cost of compromised efficiency. Searchable symmetric encryption (SSE) allows retrieval of encrypted data over cloud. For the first time, they formulated the privacy issue from the aspect of similarity relevance and scheme robustness. And observed that server-side ranking based on order-preserving encryption (OPE) inevitably leaks data privacy. To eliminate the leakage, they proposed a two-round searchable encryption (TRSE) scheme that supports top-k multikeyword retrieval. In TRSE, we employ a vector space model and homomorphic encryption. The vector space model helps to provide sufficient search accuracy, and the homomorphic encryption enables users to involve in the ranking while the majority of computing work is done on the server side by operations only on ciphertext. As a result, information leakage can be eliminated and data security is ensured. Thorough security and performance analysis show that this scheme guarantees high security and practical efficiency. The drawback of this scheme is as leakage of data is concerned they have not been able to completely overcome the problem.

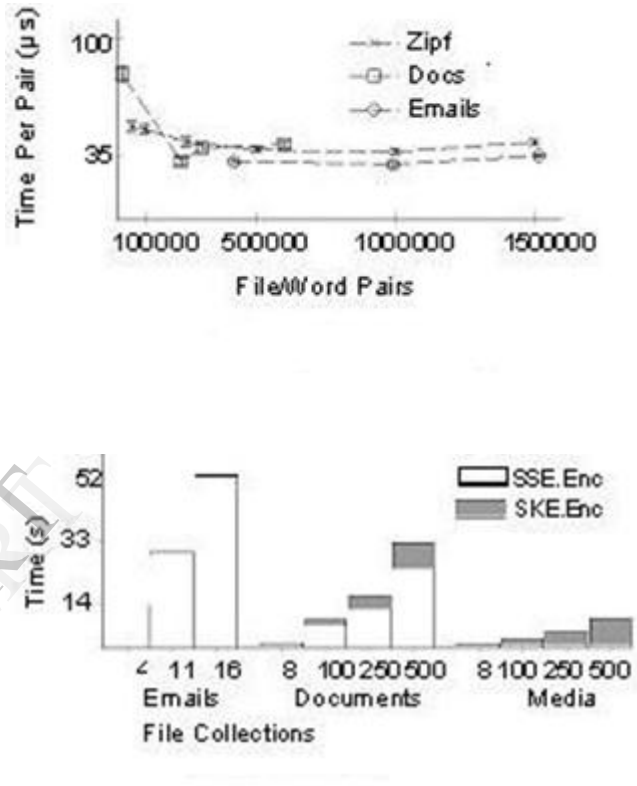

Fig 4: Graphical Representation of SSE

In [2], data owners are motivated to outsource their complex data management systems from local sites to commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in cloud, it is crucial for the search service to allow multi-keyword query and provide result similarity ranking to meet the effective data retrieval need. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely differentiate the search results.

3. System model

Cloud computing though provides a very dynamic and profitable structure; however it introduces significant concerns about privacy, security, data integrity, and intellectual property management, audit trails, and other issues. Because of the control that consumers of cloud services to providers, successful initiatives rely on a high degree of trust between a Client (Organization or university) and a supplier, including confidence in the provider's long term viability.

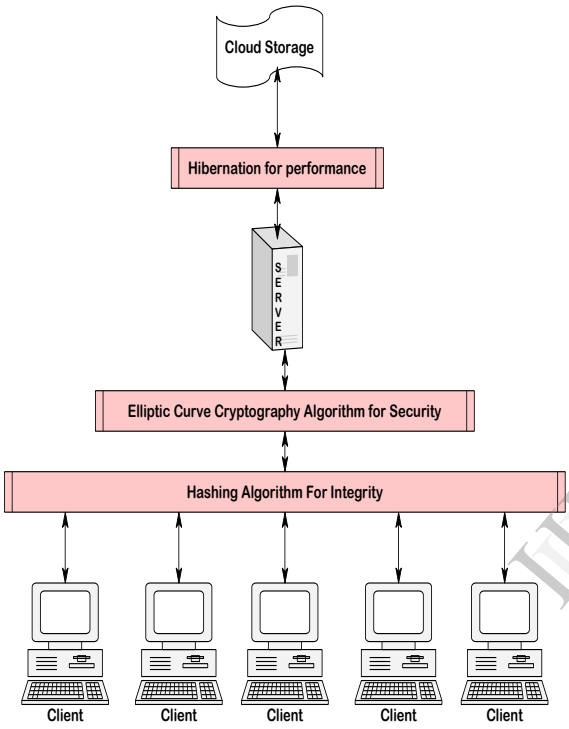

Fig 5: Cloud service architecture

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Elliptic curves are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

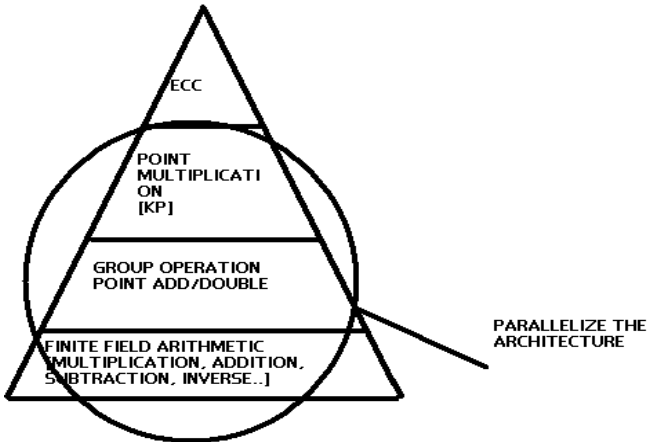

Fig 6: Ecc operations

Public-key cryptography is based on the intractability of certain mathematical problems. Early public-key systems are secure assuming that it is difficult to factor a large integer composed of two or more large prime factors. For elliptic-curve-based protocols, it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible – this is the "elliptic curve discrete logarithm problem" or ECDLP. The entire security of ECC depends on the ability to compute a point multiplication and the inability to compute the multiplicand given the original and product points. The size of the elliptic curve determines the difficulty of the problem. The primary benefit promised by ECC is a smaller key size, reducing storage and transmission requirements, i.e. that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key – e.g., a 256-bit ECC public key should provide comparable security to a 3072-bit RSA public key (see key sizes below). How we use hashing technique in ECC algorithm, this is done by applying some mathematical function to the key to generate a number in the range of record numbers.

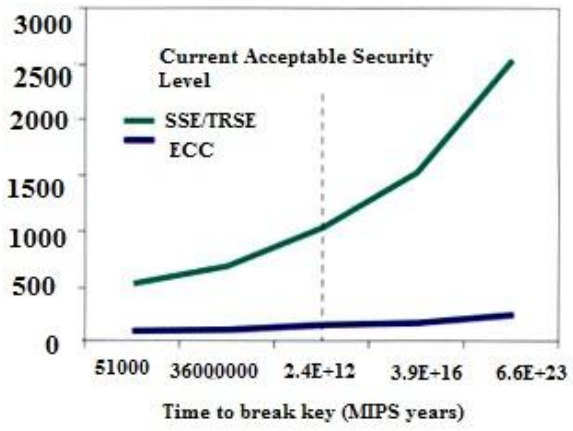

Fig 7: Ecc vs SSE/TRSE comparison graph

3.1 Security Policy Control

For the cloud computing to go main stream, it must offer IT organizations the ability to enforce corporate policy. This policy control ranges from the simple daily policy issues (like enforcing rules to ensure strong passwords) to the more complex (like conducting security related forensics). Many cloud providers fail to offer the kind of tough policy control that many organizations require. These services are provided and maintained secure by ECC and hashing technique.

3.2 Multi-tenancy Trusted Computing Model

Cloud computing is a bilateral service model in which there are two entities: CSP and customers. Customers rent for software, platform or infrastructure services from CSP.

CSP can be self-interested, untrustworthy and possibly malicious. Firstly, they are owned by CSP and organized to provide cloud services to customers, so they should be managed by CSP to satisfy the service level agreement (SLA) between CSP and customers. Secondly, as they are the platforms that customers store their data in or run their businesses on, they should supply customers with proper mechanisms to manage and secure their data or applications. In other words, they should be designed to accept and enforce the security policies from customers, which must not
be tampered by CSP or other customers. From this perspective, cloud computing should have the capability to compartmentalize each customer and CSP and support security duty separation. The key point compartmentalization and security duty separation between CSP and customers is to define clear and seamless security responsibility boundaries for CSP and customers.

4. Key Generation

From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.

Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance. Thus this is provided efficiently by ECC.

Key generation is an important part where we have to generate both public key and private key. The sender will be encrypting the message with receiver's public key and the receiver will decrypt its private key.

Now, we have to select a number **'d'** within the range of **'n'**. Using the following equation we can generate the public key

$$Q = d * P$$

d = The random number that we have selected within the range of (**1 to n-1**). **P** is the point on the curve.

'Q' is the public key and **'d' is the private key.**

4.1 Encryption

Let 'm' be the message that we are sending. We have to represent this message on the curve. This have in-depth implementation details. All the advance research on ECC is done by a company called **certicom.**

Conside *'m'* has the point *'M'* on the curve *'E'*. Randomly select 'k' from [1 - (n-1)].

Two cipher texts will be generated let it be **C1** and **C2**.

C1 = k*P
*C2 = M + k*Q*

C1 and C2 will be send.

4.2 Decryption

We have to get back the message 'm' that was send to us,

*M = C2 – d * C1*

M is the original message that we have send.

Proof

How does we get back the message,

M = C2 – d * C1

'M' can be represented as 'C2 – d * C1′

C2 – d * C1 = (M + k * Q) – d * (k * P) (C2 = M + k * Q and C1 = k * P)

= M + k * d * P – d * k *P (canceling out k * d * P)

= M (Original Message).

5.Elliptical curve Domain parameters

Apart from the curve parameters a and b, there are other parameters that must be agreed by both parties involved in secured and trusted communication using ECC. These are domain parameters. Generally the protocols implementing the ECC specify the domain parameters to be used.

For efficient implementation of ECC, it is important for the point multiplication algorithm and the underlying field arithmetic to be efficient. There are different methods for efficient implementation point multiplication and field arithmetic suited for different hardware configurations. Implementation of ECC using projective coordinates has shown considerable improvement in efficientcy compared to the affline coordinate implementation. This improvement in efficiency is due to the elimination of multiplicative inverse operation inpoint addition and doubling that would otherwise cost considerable processor cycles.

If the irreducible polynomial in binary field implementation is chosen to be trinomial or pentanomial the implementation of ECC on binary field implementation is chosen to be trinomial or pentanomial the implementation of ECC on binary field can be made efficient than the prime field implementation.

Conclusion

It is clear that although the use of cloud computing has rapidly increased; cloud computing security is still considered the major issue in the cloud computing environment. Customers do not want to lose their private information as a result of malicious insiders in the cloud. In addition, the loss of service availability has caused many problems for a large number of customers recently. Furthermore, data intrusion leads to many problems for the users of cloud computing. The purpose of this work is to provide security for both single and multi cloud environment. Implemeting ECC using projective coordinates has shown considerable improvement in efficiency compared to other cryptographic algorithms. It also provides a reduced key size. And hibernate concept is uised to provide persistent datas. And also the intruders can be identified by using Hashing technique. From these we propose a secure sharing scheme without any leakage.. Anyuser in the group can securely share data with others in the cloud. The encryption complexity and size of cipher text are independent. The proposed scheme guarantees the full efficiency.

Symmetric algorithm (bit)	**RSA and DH (bit)**	**ECC (bit)**
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1. Comparison of key size

6.References

[1] " Toward secure multikeyword top-k retrieval over encrypted cloud data" IEEE transactions on dependable secure computing vol10 no4 july2013.

[2] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS), 2010..

[3] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A.L. Varna, S. He, M.Wu, and D.W. Oard, "Confidentiality-Preserving Rank-Ordered Search," Proc. Workshop Storage Security and Survivability, 2007.

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multikeyword Ranked Search over Encrypted Cloud Data," Proc. IEEE INFOCOM, 2011

[5] C. Wang, Q. Wang, K. Ren and W. Lou, "Ensuring data storage security in CloudComputing", in Proc. OfIWQoS'09, July 2009.

[6] M. Atallah and K. Frikken, "Securely outsourcing linear algebra computations", in Procof ASIACCS, 2010, pp. 48-59.

[7] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraiccomputations," in Proc. of 6th Conf. on Privacy, Security, and Trust (PST), 2008, pp.240–245.
S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k Retrieval from a Confidential Index," Proc. 12th Int'l Conf.
Extending Database Technology: Advances in Database Technology (EDBT), 2009.

[8] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," Proc. 29th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques, H. Gilbert, pp. 24-43, 2010.

[9] M. Perc, "Evolution of the Most Common English Words and
Phrases over the Centuries," J. Royal Soc. Interface, 2012.

[10] O. Regev, "New Lattice-Based Cryptographic Constructions,"
J. ACM, vol. 51, no. 6, pp. 899-942, 2004.

[11] N. Howgrave-Graham, "Approximate Integer Common Divisors," Proc. Revised Papers from Int'l Conf. Cryptography and Lattices (CaLC' 01), pp. 51-66, 2001.

[12] "NSF Research Awards Abstracts 1990-2003," http://kdd.ics.uci. edu/databases/nsfabs/nsfawards.html, 2013.

[13] "20 Newsgroups," http://kdd.ics.uci.edu/databases/ 20newsgroups/20newsgroups.html, 2013.

[14] S. Gries, "Useful Statistics for Corpus Linguistics," A Mosaic of Corpus Linguistics: Selected Approaches, Aquilino Sanchez Moises Almela, eds., pp. 269-291, Peter Lang, 2010.

[15] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," Proc. 41st Ann. ACM Symp. Theory of computing (STOC), pp. 169-178, 2009.

[16] D. Dubin, "The Most Influential Paper Gerard Salton Never
Wrote," Library Trends, vol. 52, no. 4, pp. 748-764, 2004.
[20] A. Cuyt, V. Brevik Petersen, B. Verdonk, H. Waadeland, and W.B. Jones, Handbook of Continued Fractions for Special Functions. Springer Verlag, 2008.

[17] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, pp. 856-887. MIT Press and McGraw-Hill, 2001.

[18] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for
Searches on Encrypted Data," Proc. IEEE Symp. Security and
Privacy, 2000.

[19] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public-Key Encryption with Keyword Search," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques (Eurocrypt), 2004.

[20] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A.L. Varna, S. He, M.Wu, and D.W. Oard, "Confidentiality-Preserving Rank-Ordered Search," Proc. Workshop Storage Security and Survivability, 2007.

[21] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multikeyword Ranked Search over Encrypted Cloud Data," Proc. IEEE INFOCOM, 2011.

[26] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism," Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), 2011.

.