IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. II (July – Aug. 2015), PP 16-24
www.iosrjournals.org

Effective Modular Order Preserving Encryption on Cloud Using MHGD

N.Jayashri.[1] T.Chakravarthy.[2]
[1] *Research Scholar, AVVM Sri Pushpam College,Tamilnadu, India.*
[2] *Asso. Professor AVVM Sri Pushpam College,Tamilnadu, India.*

Abstract: *Cloud computing strengthens its presence in the public sector, Organizations and individuals are looking for cloud services to improve productivity, security and reduce costs. Apart from communication, file storage is the main requirement for common people. Traditional data centers consist of large collections of server farms implementing perimeter-security measures. Public cloud offersa multitenant service, in which the concept of the network perimeter evaporates. For the former concern, data encryption before outsourcing is the simplest way to protect data privacy. But encryption also makes deploying traditional data utilization services — a difficult. This problem on how to search encrypted data has recently gained attention and led to the development of searchable encryption techniques. In this work we are try to implement Modular Order Preserving Encryption(MOPE), a primitive which allowing a efficient modular range queries on encrypted documents. This is a kind of Searchable Encryption Scheme. MOPE improves the security of OPE in the sense, as it does not leak any information about the location of plaintext, Boldyvera et.al. Main goal of this work is to improve the security provided by the existing MOPE approaches with the help of Multivariate Hypergeometric Distribution (MHGD).*

Keywords: *Deterministic Encryption, Hypergeometric Distribution, Modular OPE, Multivariate HGD, Order Preserving Encryption, Range Queries, Searchable Encryption,.*

I. Introduction

In the evolution of computing technology, information processing has moved from mainframes to personal computers to server-centric computing to the Web. Today, many organizations are seriously considering adopting cloud computing, the next major milestone in technology and business collaboration[1]. Cloud computing has been defined by NIST(National Informatic Science and Technonology) as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or cloud provider interaction. Cloud Computing remains a work in progress [2].

Fig.1. Cloud Structure

Although cloud computing's benefits are tremendous, security and privacy concerns are the primary obstacles to wide adoption [3]. Because cloud service providers (CSPs) are separate administrative entities, moving to the commercial public cloud deprives users of direct control over the systems that manage their data and applications. Even if CSPs' infrastructure and management capabilities are much more powerful and reliable than those of personal computing devices, the cloud platform still faces both internal and external security and privacy threats, including media failures, software bugs, malware, administrator errors and malicious insiders. Noteworthy outages and security breaches to cloud services appear from time to time[3].

Because users don't have access to the cloud's internal operational details, CSPs might also voluntarily examine users' data for various reasons without detection[4].Although it increases resource utilization, this unique multitenancy feature also presents new security and privacy vulnerabilities for user interactions[5].Hence, we argue that the cloud is intrinsically insecure from a user's viewpoint. Without providing a strong security and privacy guarantee, we can't expect users to turn control of their data and computing applications over to the cloud based solely on economic savings and service flexibility[3].

According to users involved in the cloud can be classified in to three categories. A public cloud is one in which the infrastructure and other computational resources that it comprises are made available to general public over the Internet. It is owned by a cloud provider selling cloud services and by definition is external to an organization greater control over the infrastructure and computational resources than does a public cloud [2].As individuals and enterprises produce more and more data that must be stored and utilized , they're motivated to outsource their local complex data management systems to the cloud owing to its greater flexibility andcost-efficiency. However, once users no longer physically possess their data, its confidentiality and integrity can be at risk[4].Traditionally, to control the dissemination of privacy-sensitive data, users establish a trusted server to store data locally in clear, and then control that server to check whether requesting users present proper certification before letting them access the data[8]. From a security standpoint, this access control architecture is no longer applicable when we outsource data to the cloud.Data encryption before outsourcing is the simplest way to protect data privacy and combat unsolicited access in the cloud and beyond. But encryption also makes deploying traditional data utilization services — such as plaintext keyword search over textual data or query over database — a difficult task. The trivial solution of downloading all the data and decrypting it locally is clearly impractical, due to the huge bandwidth cost resulting from cloud-scale systems. Moreover, aside from eliminating local storage management, storing data in the cloud serves no purpose unless people can easily search and utilize that data.

Another important issue thatarises when outsourcing dataservice to the cloud is protecting data integrity and long-term storagecorrectness. Although outsourcingdata to the cloud is economicallyattractive for long-term, largescalestorage, it doesn't immediatelyguarantee data integrity andavailability. This problem, if notproperly addressed, can impede thesuccessful deployment of a cloudarchitecture. Given that users nolonger locally possess their data, theycan't utilize traditional cryptographicprimitives to protect its correctness[5].Such primitives usually require alocal copy of the data for integrityverification, which isn't viable whenstorage is outsourced. Furthermore,the large amount of cloud data andthe user's constrained computingcapabilities make data correctnessauditing in a cloud environmentexpensive and even formidable [5].Other challengingsecurity problems include assureddata deletion and remote assessmentof fault tolerance that is, the remotedetection of hard-drive failurevulnerabilitiesin the cloud[7].**U**ltimately, the cloud is neither good nor bad: it's just a new paradigm with its own advantages and disadvantages. Over time, some of these concerns will be solved or the risks will be reduced to acceptable levels. For now, these concerns have kept cloud adoption at a modest pace.[6]

The rest of the paper is organized as follows: Section 2 List, some of the Searchable Encryption techniques. Existing works in Order Preserving Encryption is listed in Section 3. Section 4, list the designing goals of this work .Section 5 explain about our proposed work. Performance analysis is dicussed in section 6. Section 7 presents a security analysis of our approach. Finally Section 8 gives the conclusion of the whole work done in this paper.

II. Searchable Encryption

The problem on how to search encrypted data has recently gained attention and led to the development of searchable encryption techniques. At a high level, a searchable encryption scheme employs a prebuilt encrypted search index that lets users with appropriate tokens securely search over the encrypted data via keywords without first decrypting it. However, considering the potentially large number of on-demand data users and the huge amount of outsourced data files in the cloud, this problem is still particularly challenging because meeting performance, system usability, and scalability requirements is extremely difficult. In this context, numerous interesting yet challenging problems remain, including similarity search over encrypted data, secure ranked search over encrypted data, secure multikeyword semantic search, secure range query, and even secure search over non-textual data such as graph or numerical data.

2.1. Probabilistic Encryption.

This is the encryption scheme use randomness in an encryption algorithm, so that when encrypting the same message several times it will, in general, yield different ciphertexts. The term "probabilistic encryption" is typically used in reference to public key encryption algorithms, however varioussymmetrickey encryption algorithms achieve a similar property (e.g., block ciphers when used in a chaining mode such as CBC). To be semantically secure, that is, to hide even partial information about the plaintext, an encryption algorithm must be probabilistic.Probabilistic encryption is particularly important when using public key cryptography. Suppose that the adversary observes a ciphertext, and suspects that the plaintext is either "YES" or "NO", or has a hunch that the plaintext might be "ATTACK AT CALAIS".

2.2. Deterministic encryption.

Thisis a cryptosystem which always produces the same ciphertext for a given plaintext and key, even over separate executions of the encryption algorithm. Examples of deterministic encryption algorithms include RSA cryptosystem (without encryption padding), and many block ciphers when used in ECB mode or with a constant initialization vector.When a deterministic encryption algorithm is used, the adversary can simply try encrypting each of his guesses under the recipient's public key, and compare each result to the target ciphertext. To combat this attack, public key encryption schemes must incorporate an element of randomness, ensuring that each plaintext maps into one of a large number of possible ciphertexts.An intuitive approach to converting a probabilistic encryption scheme into a deterministic one is to simply avoid padding in the plaintext before encrypting with the probabilistic algorithm.

2.3. Homomorphic Encryption.

We want to query a search engine, but don't want to tell the search engine what we are looking for? We might consider encrypting our query, but if we use an ordinary encryption scheme, the search engine will not be able to manipulate our ciphertexts to construct a meaningful response. What we would like is a cryptographic equivalent of a photograph developer's "dark room", where the search engine can process our query intelligently without ever seeing it [23].A "fully homomorphic" encryption scheme creates exactly this cryptographic dark room. Using it, anyone can manipulate ciphertexts that encrypt data under some public key 'pk' to construct a ciphertext that encrypts *any desired function* of that data under 'pk'. Such a scheme is useful in the settings above.

In 2009, Gentry proposed the first efficient fully homomorphic encryption scheme. It is efficient in the sense that all algorithms run in time polynomial in the security parameter and the size of the function f that we are computing, and the size output ciphertext grows only linearly with the size off's output. Although all algorithms run in polynomial time, there is still work to be done to make it truly practical.Members of the group are very active in investigating new forms of homomorphic encryption and also in implementations to test its practical applicability.Homomorphic encryptions allow complex mathematical operations to be performed on encrypted data without compromising the encryption. In mathematics, homomorphic describes the transformation of one data set into another while preserving relationships between elements in both sets. The term is derived from the Greek words for "same structure." Because the data in a homomorphic encryption scheme retains the same structure, identical mathematical operations ─ whether they are performed on encrypted or decrypted data ─ will yield equivalent results.Homomorphic encryption is expected to play an important part in cloud computing, allowing companies to store encrypted data in a public cloud and take advantage of the cloud provider's analytic services [24].

A breakthrough theoretical approach, fully homomorphicencryption (FHE)[12], and a number of follow-up works, while generic, is currently inefficient, and seems unlikely to become truly practical in the foreseeable future. Although a significant effort is underway in the theoretical community to improve the performance of FHE, it is unlikely that fully-homomorphic encryption will approach the efficiency of current public key encryption (PKE) schemes any time soon.[9]

III. Existing Work

Searchable encryption is still far from providing the same search usability, functionality, and flexibility as in plaintext search. How to create the same search experiences over encrypted cloud data for users, while providing the security and privacy guarantees? To enable semantic –rich encrypted search over largescale cloud data. Order Preserving Encryption(OPE) can be viewed as a tool somewhat similar to fully-homomorphic encryption, in that it can repeatedly operate on encrypted data. It is weaker than FHE since the manipulation primitive is limited to equality checking and comparisons.[9]

3.1. Order Preserving Encryption.

Much of the value of cloud services lies in leveraging client data, which often conflicts with the client's desire to keep that data private. Generic theoretical approaches, such as fully-homomorphic encryption, are inefficient. Ad hoc approaches, such as OPE, provide solutions to a limited class of problems (e.g., evaluating encrypted range queries). [9]. OPE was proposed in the database community by Agrawal[11].E is an order preserving encryption function, and p_1 and p_2 are two plaintext values,and

$c_1 = E(p_1)$.
$c_2 = E(p_2)$.
if (p1 < p2) then (c1 < c2).

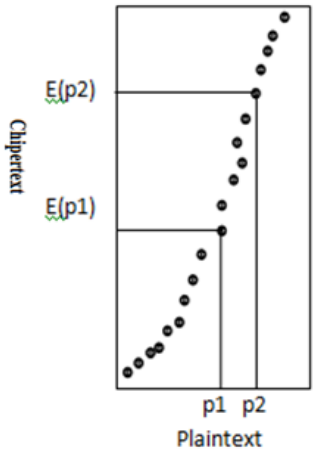

Fig. 2. Order Preserving Encryption.

c1 and c2 are two corresponding chipertext values.

Query results from OPE will be sound and complete. Comparison operations will be performed without decrypting the operands. It also Tolerate updates [10].OPE Encryption is a two-step process. First step is, Source (plaintext) to uniform conversion. And the second step is, Uniform to target (ciphertext) conversion. For Decryption, above steps are performed in a reverse order.

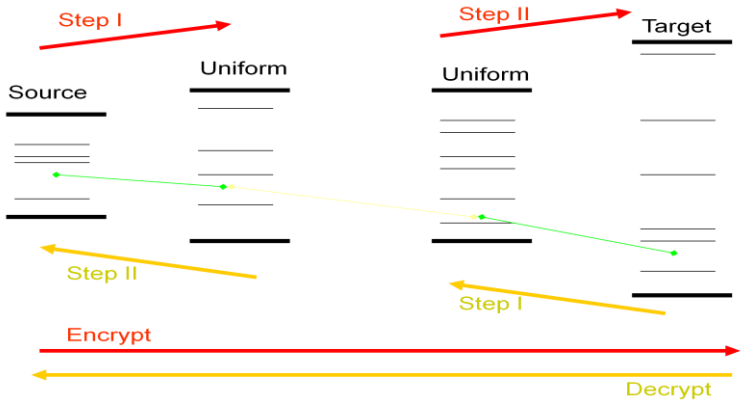

Fig.3. Steps Involved in OPE.

There are a number of applications which could benefit from order-preserving encryption. For privacy protection the word frequency values are encrypted with OPE, enabling a ranked search on the indexes. Wang et al. [13] propose a scheme that supports secure and efficient ranked keyword searches over encrypted datastored in the cloud by applying order-preserving encryption on certain relevance criteria such as the frequency of keywords.

Ding and Klein [14] propose an application-level encryption solution to protect the privacy and confidentiality of health data. In particular, their solution relies on order-preserving encryption to enable some operation on dates expressed in milliseconds without first having to decrypt them. These and other applications of OPE (e.g., [15, 16, and 17]) all target an outsourced computation or storage model, which are key characteristics of cloud computing. Order-preserving encryption is often seen as a powerful cryptographic tool that can be securely plugged into existing systems.[9].OPE algorithm following three steps: modeling the input

and target distributions, attening the plaintext database into a at database, and transforming the at database into the cipher database.

3.2. Order Preserving Symmetric Encryption.

An order-preserving symmetric encryption scheme is a deterministic symmetric encryption scheme whose encryption algorithm produces ciphertexts that preserve numerical ordering of the plaintexts. In fact, OPE not only allows efficient range queries, but allows indexing and query processing to bedone exactly and as efficiently as for unencrypted data, since a query just consists of the encryptionsof a and b and the server can locate the desired ciphertexts in logarithmic-time via standard tree-baseddata structures[18].Allowing range queries on encrypted data in the public-key setting was studied in [19, 20]. While their schemes provably provide strong security, they are not efficient, requiring to scan the whole database on every query.

We turn to an approach along the lines of pseudorandom functions (PRFs) or permutations (PRPs), requiring that no adversary can distinguish between oracle access to the encryption algorithm of the scheme or a corresponding "ideal" object. In our case the latter is a random order-preserving [18]. Blockciphers,usual tool in the symmetric-key setting, do not seem helpful in preserving plaintext order. Construction proposed by A Boldyreva et. al.[18] takes a different route, borrowing some tools from probability theory. They uncover a relation between a random order-preserving function and the hypergeometric (HG) and negative hypergeometric (NHG) probability distributions.First, assigning multiple plaintexts to ciphertexts independently accordingto the NHG distribution cannot work, it require frequent adjustment in the parameters of the NHG sampling algorithm appropriately for each new plaintext. But we want astateless scheme. Instead of making the long random tape the secret key K for our scheme, we can makeit the key for a PRF and generate portions of the tape dynamically as needed. Since the size of parameters to the NHG sampling algorithm as well as the number of random coins it needs varies during the binary search, and also because such a construction seemsuseful in general. Finally, our scheme needs an efficient sampling algorithm for theNHG distribution. We turn to a related probability distribution, namely the hypergeometric (HG) distribution,for which a very efficient exact sampling algorithm [18].

A Boldyvera et. al.[18] address the open problem of characterizing what encryption via a random order-preserving function (ROPF) leaks about underlying data. In particular, they show that, for a database of randomly distributed plaintexts and appropriate choice of parameters, ROPF encryption leaks neither the precise value of any plaintext nor the precise distance between any two of them. On the other hand, they show that ROPF encryption leaks approximate value of any plaintext as well as approximate distance between any two plaintexts, each to an accuracy of about square root of the domain size. They also study schemes that are not order-preserving, but which nevertheless allow efficient range queries and achieve security notions stronger than Psuedorandom Order Preserving Encryption(POPF)[18]. In a setting where the entire database is known in advance of key-generation, we show that recent constructions of monotone minimal perfect hash functions allow to efficiently achieve the notion of IND-OCPA(Indisdiguisability-Chosen Plaintext Attack) also considered by Boldyreva et al., which asks that only the order relations among the plaintexts is leaked.

IV. Proposed Method

4.1. Using Modular Order Preserving Encryption

Modular order-preserving encryption (MOPE),due to Boldyreva et al. [8], is a promising extension thatincreases the security of the basic OPE by introducing a secretmodular offset to each data value prior to encryptingit. However, executing range queries via MOPE in a nativeway allows the adversary to learn this offset, negating anypotential security gains of this approach. We try to implement modular order-preserving encryption (MOPE), in which the scheme of Multivariate Hypergeometric Distribution (MHGD) prepended with a OPE. MOPE with MHGD improves the efficiency of MOPE in a sense, as it Produce coins which are more complicated to brute force.

A modular order-preserving encryption (MOPE) scheme is an extension to OPE that increases its security. Instead of defining such a scheme in general, we define a transformation to obtain it from a given OPE scheme.The transformation. Let OPE = (Kg´; Enc´; Dec´) be an OPE scheme. We define the associated modular OPE scheme MOPE[OPE] = (Kg; Enc; Dec) where

Notations used

Kg = Key generator

Enc = Encryption Algorithm

Dec =Decryption Algorithm

M = Group size

D = Sub_group size

n =Sample size

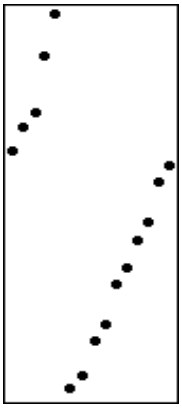

Fig. 4.MOPE (Encrypted Value Distribution).

* Kg generates K ←$ Kg′ and j← $[M]; it outputs (K; j).
* Enc on inputs a key K and a plaintext m outputs Enc′ (K, m + j mod M).
* Dec on inputs a key K and a ciphertext c outputs Dec(K; c) ─ j mod M.

Above, the value j in the secret key of MOPE[OPE] is calledthe secret offset or displacement.

4.2. Using Multivariate Hypergeometric Distribution

Discrete distributions can only take a discrete number of values. This number may be infiniteor finite. In HGD, Models the number of items of a particular type there will be in a sample of size n where that sample is drawn from a population of size 'M' of which 'D' are also of that particular type. An extension of the Hypergeometric distribution where more than two sub-populations of interest exist is called Multivariate Hypergeometric distribution.Multivariate distributions describe several parameters whose values are probabilistically linked in some way[23]. The MHGD is created by extending the mathematics of the HGD. For the HGD with a sample of size n, the probability of observing s individuals from a sub-group of size M, and therefore (n-s) from the remaining number (M-D):

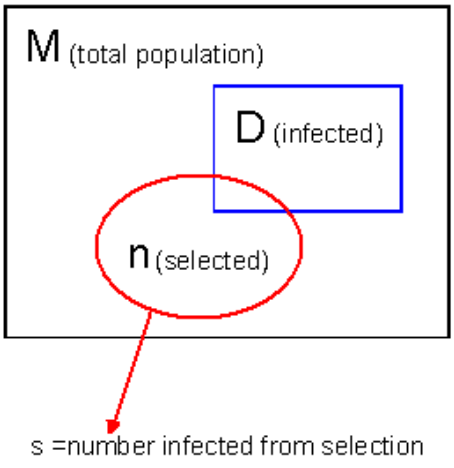

Fig.5. Multivariate Hypergeometric distribution

And results in the probability distribution for s:

$$f(x) = \frac{\binom{D}{x}\binom{M-D}{n-x}}{\binom{M}{n}}$$

The numerator is the number of different sampling combinations (each of which has the same probability because each individual has the same probability of being sampled) where one would have exactly s from the sub-group D (and by implication (n-s) from the sub-group (M-D). The denominator is the total number of different combinations of individuals one could have in selecting n individuals from a group of size M. Thus the equation is just the proportion of different possible scenarios, each of which has the same probability, that would give us s from D [23]. The Multivariate Hypergeometric probability equation is just an extension of this idea. D_1, D_2, D_3 and so on are the number of individuals of different types in a population, and x_1, x_2, x_3, ... are the number of successes. And results in the probability distribution for {s}:

$$f(x) = \frac{\binom{D_1}{x_1}\binom{D_2}{x_2}\cdots\binom{D_k}{x_k}}{\binom{M}{n}}$$

where $$\sum_{i=1}^{k} D_i = M, \sum_{i=1}^{k} x_i = n$$

5.2. Psudocode for Encryption Algorithm

Existing MOPE method, use a HGD method for coin generation. We alter that in a simple way to using MHGD method for coin generation. Below mentioning psudocode describe the notations and logic which are used to implement MHGD in MOPE. See encryption algorithm for the formal descriptions of Enc, where as before $l_1 = l(D,R,y)$ is the number of coins needed by MHGD on inputs D,R, y, and l_R is the number of coins needed to select an element of R uniformly at random.

Encryption Algorithm for Using MHGD for MOPE

$Encryption_{Key}(D,R,m)$

1. Assign |D| to M and |R| to N.
2. Calculate min(D)-1 and assign it to d;
3. Calculate min(R)-1 and assign it to r;
4. Calculate [N/2], add with 2 and assign it toy;
5. Check whether |D| = 1 then
   a. Invoke TapeGen function with parameters $K,1^{l1}$,(D,R,0||Y)) assign the result to cc.
   b. Assign R to c.
   c. Return c.
6. Return Encrypted values.

Algorithm for Tapegeneration

1. Calculate MHGD with parameters D,R,y,n;cc and assign the result to x.
2. Check If m is less than are equal to x then
   a. Assign {d+1,.....x} to D.
   b. Assign {r+1,.....y} to R.
3. Else
   a. Assign {x+1,.....,d+M} to D.
   b. Assign {y+1,.....,r+N} to R.

The efficiency of our scheme follows from our previous analyses. Encryption and decryption require the time for at most logN + 3 invocations of MHGD on inputs of size at most logN plus at most (5 logM + 14) . (5 logN + $\lambda^{'}$ + 1)=128 invocations of AES on average for $\lambda^{'}$ in the theorem.

5. Security Analysis

We show that a random modular OPF, unlike a random OPF, completely hides the locations of the data points.We will also try to sort out leakage with respect to distance and window-distance one-wayness. On the

other hand, if the adversary is able to recover a single known plaintext-ciphertext pair, security falls back to that of a random OPF in Previous Scheme but our Proposed method not exactly reveal the plaintext - chipertext pair.

We propose a changes to an existing MOPE scheme that also improves the security performance of any OPE. The resulting scheme is no longer strictly order-preserving, but it still permits range queries. However, now the queries must be modular range queries. Standard range queries are not supported, as only "modular order" rather than order is leaked. The changes in MOPE is simple, generic, and basically free computation-wise.

Notice that a MOPE is suitable for modular range query support as follows. To request the ciphertexts of the messages in the range [m1;m2] (if m1 _ m2), or [m1;M][[1;m2] (if m1 > m2), the user computes c1 Encm(K;m1); c2 Encm(K;m2) and submits ciphertexts (c1; c2) as the query. The server returns the ciphertexts in the interval [c1; c2] (if c1 _ c2) or [c1;N] [[1; c2] (if c1 > c2).Note that an MOPF could alternatively be defined with a MHGD following the OPF rather than a random plaintext shift preceding it. The advantage of the above definition is that the map from (OPF, ciphertext offset) pairs to MOPFs is bijective whereas in the alternative it is not one-to-one.

5.1. Performance Analysis

We propose a technique that improveson the efficiency of any MOPE scheme without sacrificing security. ROPF analysisreveals information leakage in OPE not alluded to by [9], namely about the locations of the data pointsrather than just the distances between them. We suggest a modification to an MOPE scheme thatovercomes this. The modification to the scheme issimple and generic: the encryption algorithm just adds a secretoffset to the message before encryption. The secret offset is the same for all messages. We use amethodMHGD for modular OPE scheme, and generalize the security notion: the ideal objectis now a random modular OPF (RMOPF), i.e. a random OPF applied to messages with a randomlypicked offset. It is easy to see that any MOPE scheme, using MHGD yields a efficient architecture for the above transformation.

V. Conclusion

We revisited security of symmetric order-preserving schemes defined in [9]. We formally clarify thestrengths and limitations of any OPE scheme proven to be a pseudorandom order-preserving function(POPF), and in particular, the efficient OPE scheme proposed in [9]. Namely, for any POPF-secure OPEour analysis together with the result of [9] provides upper bounds on the advantages of any adversariesattacking the one-wayness and distance one-wayness, (2) lower bounds on the window one-wayness andwindow distance one-wayness advantages. We hope our results help practitioners to estimate the risksand security guarantees of using a secure OPE in their applications. Our analysis also gives directionsin selecting the size of the ciphertext space. Finally we propose a simple and efficient transformationthat can be applied to any MOPE scheme. Our analysis shows that the transformation yields a schemewith improved efficiency in that the scheme resists the one-wayness and window one-wayness attacks.

References

[1]. Crowe Horwath LLP, Warren Chan, Eugene Leung,Heidi Pili. Enterprise Risk Management for Cloud Computing.ResearchCommissionedby COSO.(June 2012).

[2]. Wayne Jansen Timothy Grance. NIST-Draft-SP-800-144_cloud-computing - Guidelines on Security and Privacy in Public Cloud Computing.

[3]. "Security Guidance for Critical Areas of Focus in Cloud Computing," Cloud Security Alliance, Dec. 2009; https://cloudsecurityalliance.org/csaguide.pdf.

[4]. Kui Ren, Cong Wang, and Qian Wang . Security Challenges for the Public Cloud. Illinois Institute ofTechnology.

[5]. C. Wang et al., Privacy-Preserving Public Auditing for Storage Security in Cloud Computing, Proc. 30th IEEE Int'l Conf. Computer Communications (INFOCOM 10), IEEE Press, 2010, pp. 525–533.

[6]. Paul Hofmann .Cloud Computing: The Limits of Public Clouds for Business Applications.SAP Labs

[7]. Dan Woods.CITO Research.

[8]. K. Bowers et al. How to Tell if Your Cloud Files Are Vulnerable to Drive Crashes.Proc. 18th ACM Conf. Computer and Communications Security (CCS 11), ACM Press, 2011, pp. 501–514.

[9]. S. Yu et al.Achieving Secure, Scalable, and Fine-Grained Access Control in Cloud Computing, Proc. 30th IEEE Int'l Conf. Computer Communications (INFOCOM 10), IEEE Press, 2010, pp. 534–542.

[10]. Vladimir Kolesnikov and Abdullatif Shikfa.On The Limits of Privacy Provided by Order- Preserving Encryption. Bell Labs Technical Journal.

[11]. Boldyreva, N. Chenette, and A. O'Neill. Order-preserving encryption revisited: improved:security analysis and alternative solutions. In Proceedings of the 31st International Conference on Advances in Cryptology, CRYPTO, 2011.

[12]. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In Proceedingsof the ACM International Conference on Management of Data, SIGMOD, 2004.

[13]. C. Gentry.Fully Homomorphic EncryptionUsing Ideal Lattices. Proc. 41st ACM Symp. OnTheory of Comput. (STOC '09) (Bethesda, MD,2009), pp. 169–178.

[14]. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou. Secure Ranked Keyword Search overEncrypted Cloud Data. Proc. 30th IEEEInternat. Conf. on Distrib. Comput. Syst.(ICDCS '10) (Genova, Ita., 2010), pp. 253–262.

[15]. Y. Ding and K. Klein.Model-DrivenApplication-Level Encryption for the Privacy ofE-Health Data,Proc. 5th Internat. Conf. OnAvailability, Reliability, and Security (ARES '10)(Krakow, Pol., 2010), pp. 341–346.

[16]. H. Liu, H. Wang, and Y. Chen, Ensuring DataStorage Security Against Frequency-BasedAttacks in Wireless Networks. Proc. 6th IEEEInternat. Conf. on Distrib. Comput. in SensorSyst. (DCOSS '10) (Santa Barbara, CA, 2010),LNCS vol. 6131, pp. 201–215.
[17]. R. A. Popa, C. M. S. Redfield, N. Zeldovich, andH. Balakrishnan. CryptDB: ProtectingConfidentiality with Encrypted QueryProcessing. Proc. 23rd ACM Symp. OnOperating Syst. Principles (SOSP '11)(Cascais, Prt., 2011), pp. 85–100.
[18]. Q. Tang. Privacy Preserving Mapping SchemesSupporting Comparison. Proc. ACM CloudComput. Security Workshop (CCSW '10)(Chicago, IL, 2010), pp. 53–58.
[19]. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In Proceedings of the 28th International Conference on Advances in Cryptology, EUROCRYPT, 2009.
[20]. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In Proceedings of the
[21]. 4th Theory of Cryptography Conference, TCC, 2007
[22]. E. Shi, J. Bethencourt, T-H. H. Chan, D. Song, and A. Perrig. Multi-dimensional range queryover encrypted data. In Symposium on Security and Privacy '07, pp. 350{364. IEEE, 2007.
[23]. V. Kachitvichyanukul and B. W. Schmeiser. Computer generation of hypergeometric randomvariates. Journal of Statistical Computation and Simulation, 22(2):127-145, 1985.
[24]. A. J. Walker. An efficient method for generating discrete random variables with general distributions.
[25]. ACM Transactions on Mathematical Software, 3:253-256, 1977.
[26]. Discrete distributions. http://www.vosesoftware.com/index.php.
[27]. Homomorphic Encryption. http://www.wikipedia.com/Homomorphic Encryption.php.