Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data

Narendrakumar Borse[1], Swapnil Mahale[2], Ujjwal Mahale[3], Kailash Panigrahi[4], Yograj Shisode[5]

Department of computer Engineering
Matoshri college of Engineering and Research centre, Eklahare, Nashik.

***Abstract*— The cloud computing is a utility that leases out the computing and storage capacities to the public individuals. In such a framework, the individual user can remotely store her data on the cloud server, namely data outsourcing, and then make the cloud data open for public access through the cloud server. This represents a more scalable, low-cost and stable way for public data access because of the scalability and high efficiency of cloud servers. Outsourced data may contain sensitive privacy information. It is often necessary to encrypt the private data before transmitting the data to the cloud servers. The data encryption, how-ever, would significantly lower the usability of data due to the dificulty of searching over the encrypted data. To address the above issues, we develop the searchable encryption to enable searching over encrypted cloud data.**

***Keywords*— CUDA, GPU, Pattern Matching, Parallel Algorithm**

I. INTRODUCTION

Privacy-preserving multi-keyword ranked search over encrypted cloud data (MRSE), and establish a set of strict privacy requirements for such a secure cloud data utilization system to become a reality. Among various multi-keyword semantics, we choose the efficient principle of coordinate matching, i.e., as many matches as possible, to capture the similarity between search query and data documents, and further use inner product similarity to quantitatively formalize such principle for similarity measurement. FTM is an innovative perspective on creating and managing fault tolerance that shades the implementation details of the reliability techniques from the users by means of a dedicated service layer. This allows users to specify and apply the desired level of fault tolerance without requiring any knowledge about its implementation.

II. LITERATURE SURVEY

This chapter highlights the succinct research contributions in developing system for multi-touch system for multiple user .Researchers have proposed a variety of applications for supporting multi-touch environment by gesture recognition but still it is only able to operate for single screen to single user. This section reviews these pro- posed techniques and summarizes the advantages and disadvantages associated with each technique.

a) Algorithms

1. RSA Algorithm

RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adelman, who first publicly described it in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it was classified until 1997. A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message. Whether breaking RSA encryption is as hard as factoring is an open question known as the RSA problem.

a)Operation

The RSA algorithm involves three steps: key generation, encryption and decryption.

b) Key generation

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q.
   - For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primarily test.
2. Compute n = pq.
   - n is used as the modulus for both the public and private keys
3. Compute $\varphi(n) = (p-1)(q-1)$, where φ is Euler's totient function.
4. Choose an integer e such that $1 < e < \varphi(n)$ and greatest common divisor of $(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are co-prime.
   - e is released as the public key exponent.
   - *e* having a short bit-length and small Hamming weight results in more efficient encryption - most commonly

0x10001 = 65,537. However, small values of *e* (such as 3) have been shown to be less secure in some settings.

c) Encryption

Alice transmits her public key (n, e) to Bob and keeps the private key secret. Bob then wishes to send message **M** to Alice.

He first turns **M** into an integer m, such that $0 < m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text c corresponding to

$$c = m^e \pmod{n}$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice. Note that at least nine values of m could yield a cipher text c equal to m, but this is very unlikely to occur in practice.

d) Decryption

Alice can recover m from c by using her private key exponent d via computing

$$m = c^d \pmod{n}.$$

Given m, she can recover the original message **M** by reversing the padding scheme.(In practice, there are more efficient methods of calculating c^d using the pre computed values below.)

2) **K-Nearest Neighbor**

K-nearest neighbor search identifies the top *k* nearest neighbors to the query. This technique is commonly used in predictive analytics to estimate or classify a point based on the consensus of its neighbors. *K-nearest* neighbor graphs are graphs in which every point is connected to its *k* nearest neighbors.

The basic idea of our new algorithm: The value of d_{max} is decreased keeping step with the ongoing exact evaluation of the object similarity distance for the candidates. At the end of the step by step refinement, d_{max} reaches the optimal query range Ed and prevents the method from producing more candidates than necessary thus fulfilling the r-optimality criterion.

NearestNeighhorSearch (q, k) // optimal algorithm

1. Initialize ranking = index.increm-ranking (F(q), df)
2. Initialize result = new sorted-list (key, object)
3. Initialize d_{max} = w
4. While o = ranking.getnext and d,(o, q) I d,,, do
5. If do@, s> s d_{max} then result.insert (d,(o, q) , o)
6. If result.length 2 k then d_{max} = result[k].key
7. Remove all entries from result where key > d_{max}
8. End while

Report all entries from result where key I d_{max}

III. PROPOSED SYSTEM

Multi-keyword ranked search over encrypted cloud data (MRSE) while preserving strict system-wise privacy in cloud computing paradigm. Among various multi-keyword semantics, we choose the efficient principle of "coordinate matching", i.e., as many matches as possible, to capture the similarity between search query and data documents. Specifically, we use "inner product similarity", i.e., the number of query keywords appearing in a document, to quantitatively evaluate the similarity of that document to the search query in "coordinate matching" principle. During index construction, each document is associated with a binary vector as a sub-index where each bit represents whether corresponding keyword is contained in the document. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by inner product of query vector with data vector. However, directly outsourcing data vector or query vector will violate index privacy or search privacy. To meet the challenge of supporting such multi-keyword semantic without privacy breaches, we propose a basic MRSE scheme using secure inner product computation, which is adapted from a secure *k*-nearest neighbor (*kNN*) technique, and then improve it step by step to achieve various privacy requirements in two levels of threat models.

IV. 1) Explore the problem of multi-keyword ranked search over encrypted cloud data, and establish a set of strict privacy requirements for such a secure cloud data utilization system to become a reality.

V. 2) Propose two MRSE schemes following the principle of "coordinate matching" while meeting different privacy requirements in two levels of threat models.

VI. 3) Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given, and experiments on the real-world dataset further show proposed schemes indeed introduce low overhead on computation and communication.

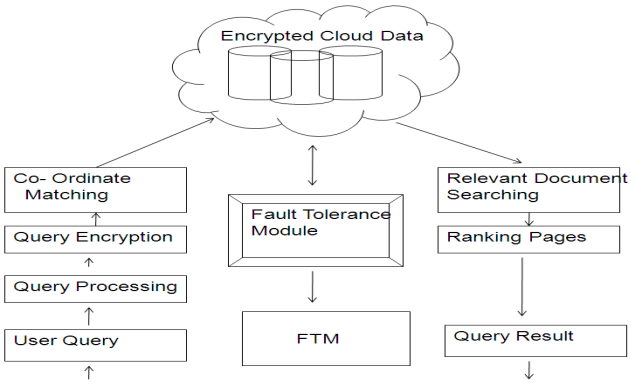

Figure 1 System Architecture

References

[1] Black Book .NET Framework 4.0

[2] Beginning of .NET 2008

[3] G. Devi, M. Pramod Kumar, "*Cloud Computing: A CRM Service Based on a Separate Encryption and Decryption using Blowfish algorithm*" International Journal of Computer Trends and Technology- volume3 Issue 4, 2012

[4] Nilesh Kumbhar, Virendra Chaudhari, Mohit Badhe, "*The Comprehensive Approach for Data Security in Cloud Computing*": A survey- International Journal of computer Applications (0975-8887) Volume 39-N0.18, Feb 2012

[5] Saranya Iswaran and Dr.Sunitha Abburu, "*Identifying Data Integrity in the Cloud Storage- IJCSI International Journal of Computer Science Issues*", Vol. 9, Issue 2, No 1, March 2012

[6] Sravan Kumar. R," Data Integrity Proofs in Cloud Storage- IEEE paper- 2011", 4-8 Jan. 2011

[7] Wei Lu, Kenneth Chiu Yinfei Pa "*A Parallel Approach to XML Parsing*", Department of computer Science, state University of New York- Binghamton, US.

[8] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "*Secure ranked keywordsearch over encrypted cloud data*".

[9] E. Shen, E. Shi, and B. Waters, "*Predicate privacy in encryption systems*" .