Secure Data Sharing in Public Cloud

Venkataramana.K, Naveen Kumar R, Sandhya Tatekalva, Padnavathamamma M
Research Scholars, Research Supervisor
Dept. of Computer Science, S.V.University – Tirupati, A.P, India

ABSTRACT:

Secure multi-party protocols have been proposed for entities (organizations or individuals) that don’t fully trust each other to share sensitive information. Many types of entities need to collect, analyze, and disseminate data rapidly and accurately, without exposing sensitive information to unauthorized or untrusted parties. Solutions based on secure multi-party computation guarantee privacy and correctness, at an extra communication (too costly in communication to be practical) and computation cost. The high overhead motivates us to extend this SMC to cloud environment which provides large computation and communication capacity which makes SMC to be used between multiple clouds (i.e., it may between private or public or hybrid clouds). Cloud may encompass many high capacity servers which act as a hosts which participate in computation (IaaS and PaaS) for final result, which is controlled by Cloud Trusted Authority (CTA) for secret sharing within the cloud. The communication between two clouds is controlled by High Level Trusted Authority (HLTA) which is one of the hosts in a cloud which provides MgaaS (Management as a Service). Due to high risk for security in clouds, HLTA generates and distributes public keys and private keys by using Carmichael-R-Prime-RSA algorithm for exchange of private data in SMC between itself and clouds. In cloud, CTA creates Group key for Secure communication between the hosts in cloud based on keys sent by HLTA for exchange of Intermediate values and shares for computation of final result. Since this scheme is extended to be used in clouds (due to high availability and scalability to increase computation power) it is possible to implement SMC practically for privacy preserving in data mining at low cost for the clients.

Keywords: Privacy preserving- Cloud-Computing - secure multi-host communication in cloud - Carmichael-R-Prime-RSA.

1. INTRODUCTION:

Cloud computing is a new type of computing, which enables convenient, on-demand access to computing resources. It is concerned with computing and storing data in the cloud, a.k.a., the Internet, where high performance machines are linked by high bandwidth connections and all the services are carefully managed and provided. It is a paradigm adopted by an increasing population with dynamic computing requirements. According to the research firm Gartner [1], Software-as-a-Service (SaaS) – one type of Cloud-based service grew 18% from $6.4 billion in 2008 to $7.7 billion in 2009, and this trend will persist through 2013. Typical examples of cloud computing include office applications migrated to the Internet such as Google Docs and enterprise computing & storage service such as Amazon EC2 & S3. Cloud computing creates a large number of security issues and challenges. A list of security threats to cloud computing is presented in [2]. Already various companies are looking forward to adopt cloud services for their business activities so every customer expects high security and privacy for its data when it is shared with other customers without revealing entire data but only what he requires. Privacy concerns arise whenever sensitive data is outsourced or accessed to/from the cloud. By using encryption, the cloud server (i.e. its administrator) is prevented from learning content in the outsourced databases. But how can we also prevent a local administrator from learning the database content. And how can we avoid scenarios such as: employees using cloud applications may learn more than it is necessary to perform their respective duties? Security practitioners, consultants and analysts interviewed for this story say cloud security vendors and cloud services providers have a long way to go before enterprise customers will be able to find a comfort zone in the public cloud, or even in a public/private hybrid deployment.

2. RELATED WORK:
A Group communication application can use multicast to transmit data all group members using minimum resource. A primary method to secure the group communication is distribution of keys used to encrypt group information. Secret sharing scheme was first proposed by Shamir [3], which can protect the secrecy of important data by distributing the data over different locations. A dealer is required to distribute the shares in Shamir’s scheme. A dealer may produce incorrect shares to other members, from which members will not be able to reconstruct a correct secret. To prevent such malicious behavior, some Verifiable Secret Sharing (VSS) schemes were suggested [4,5,6], in which inconsistent shares can be found by the receivers. The areas of VSS sharing have been studied extensively in recent years [7]. A lot of authentication schemes have been designed. Certificate-based scheme. Self-certified public key scheme was proposed by Girault and improved by Laith. G.Horng and C.S.Yang suggested an authentication scheme based on discrete logarithms in which certificate is generated by user and no extra authorities are required, whereas the authentication process is dependent on a password table. All the above schemes are practically difficult to achieve unless each member is highly efficient and supports high speed computations. So in this paper we extend the Secret sharing scheme to cloud which can provide as much computing power as required to make it practically possible to implement.

3. PROPOSED SYSTEM MODEL:

In the SDSPC we consider different clouds and each cloud may contain n hosts or parties P1, P2, ..., Pn. To protect secret data, we use Shamir’s secret sharing[3] scheme along with Carmichael-R-Prime-RSA method to protect and share the secret data among the n parties. Secure MPC algorithms consider n parties; each host has some private input. All hosts collaboratively compute a function f, which is defined on the private inputs of all the n hosts. During computation, no information about one host’s private input should be revealed to others. By computing directly on shared data, it protects the secret data even at computation time. However, existing secure computation approaches incur too high a communication overhead. It costs O(n^2) messages to perform an operation. In our model we assume that there will be High Level Trusted Authority (HLTA) for all the clouds and each cloud has a Cloud Trusted Authority (CTA) chosen by HLTA depending upon its honest history based on auditing. All the hosts will exchange their secret share with CTA and it will in turn send and receive secret share from other hosts. As the more data exchange is between only the CTA and its hosts, it reduces the communication overhead which is O(n^2) to O(n^2)/G where G is no of clouds that participate in computation. Finally the verification is done by HLTA with CTA and the output is send to users who request service from cloud. The system model is given in fig. 1. We propose this scheme at SaaS in cloud.

4. SECURITY OF CARMICHAEL-RPRIME RSA:

The security of RPrime RSA depends on the security offered by the private exponent d(as in Rebalanced RSA), and on the size of the primes employed (as in MPrime RSA) is large as it is used in cloud computing which offer high computing power. The private exponent d is large enough that attacks on small d are ineffective. Attacks on small public exponent’s e are not a problem either, due to the large e’s produced by the key generation procedure. For k = 3 and using exponents dp1, dp2 and dp3 of 160 bits each, the complexity of factoring n is O(280) using the attack of [Boneh and Shacham 2002, Wiener 1990]. To prevent factorization by ECM, primes larger than 256 bits must be employed, hence a modulus of 1024 bits can have no more than three prime factors, while a 768-bit modulus should employ at most two factors for the same reason. M. Jason Hinek analyzed a partial key exposure attack on MPrime RSA, concluding that the attack is ineffective for three and four primes. He also obtained experimental evidence that the attack has running time exponential in the size of the modulus n, which we believe can be extended for the security of Rprime RSA. We can use primes larger than 1024 bits as more computing power is available so the above Security system is feasible.

5. WORKING OF PROPOSED SCHEME:

5.1. Initialization Phase

Cloud which provides MgAS (Management As a Service) contains HLTA which generates and distributes public keys and private keys for cloud (CTA) for exchange of private data in SMC in a cloud. In this Phase

5.1.1. HLTA accepts Credentials for each cloud C, such as Cid, Cname, Cloc, CReg , CSertype ,CStorage ,CSerPay, CSerExpiry, CDomain, CSecurityParams etc.,
5.1.2. The credentials taken in above step is alphanumeric in nature, by using UNICODE scheme obtain numerical value $N_i$ for each cloud $C_i$.

5.2. Generation & Distribution of Secret Keys

Algorithm for generating Keys for exchange of data between cloud (CTA) and HLTA

5.2.1. Generate Prime factors for ‘$N_i$’ such that $N_i = \prod_{l=1}^{k} P_l$

Select $d_{pk}$ as S-bit integers such that $d_{p1}, d_{p2}, \ldots, d_{pk}$ such that

\[ \gcd(d_{p1}, p_{f1} - 1) = \ldots = \gcd(d_{pk}, p_{fk} - 1) = 1 \quad \text{and} \quad d_{p1} \equiv \ldots \equiv d_{pk} \pmod{2} \]

5.2.2. For each cloud HLTA generates a public key $e_i$ and a private key $d_i$ which is obtained from $d_{i1}, d_{i2}$ by using Carmichael -RPrime-RSA, where $d_i = d_{i1} \cdot d_{i2}$

5.2.3. Compute $d_i \equiv d_{pi} \pmod{p_i - 1}$ for $1 \leq i \leq k$ and $e_i = d_i^{-1} \pmod{\lambda(n)}$ (Public key)

5.2.4. $e_i$ is public key to each cloud, part of the private key $d_i$, i.e., $d_{i1}$ is distributed to cloud and $d_{i2}$ is retained with HLTA to exchange secure data between cloud and HLTA

5.2.5. $e_i$ is used for sending encrypted Secret Share value $S_i$ to Cloud $C_i$ by HLTA and cloud $C_i$ will decrypt by using $d_{i1}$ to get $S_i$ and $S_i$ is distributed between various hosts in $C_i$ to be used as secret Share value in polynomial.

5.2.6. Verification phase by HLTA given below.

5.3. Distribution phase of secret value between Cloud $C_i$ (CTA) and hosts in it.

The cloud $C_i$ (CTA) decrypts a secret share value $S_i$ sent by HLTA by using its private key $d_{i1}$. Then the secret $S_i$ ($S_{i1}, S_{i2}, \ldots, S_{iM}$) is sent to the hosts securely by using the following algorithm,

5.3.1. CTA selects a random number which is a Carmichael number $x_i \in \mathbb{Z}_{\lambda(N_i)}$ for each host.

5.3.2. Let $ID_i$ be the unique ID to identify each host $H_i$ in Cloud $C_i$ such that $(d_{i1} + ID_i)$ is relatively prime to $\lambda(N_i)$ where $d_{i1}$ is private key sent by HLTA to CTA.

5.3.3. The public key for all the host in cloud is $y_i = x_i / (d_{i1} + ID_i) \pmod{\lambda(N_i)}$

5.3.4. The private key is for each host is given by $K_i = a x_i \cdot d_{i1} \pmod{N_i}$ where $M$ denotes maximum possible number of hosts in a cloud and $a = \sum_{i=1}^{2^n} r_i$ random values shared by all the hosts SMC

5.3.5. The encryption and decryption is done as same in RSA between hosts and CTA.

5.3.6. Group key[18] for all host is generated as $K_M = a^{ID} \pmod{N_i}$ Where $ID_i$ is the unique ID for each in the cloud.

5.3.7. CTA sends $S_i$ to host $H_i$ by encrypting with the public key $y_i$

5.3.8. Each Host $H_i$ decrypts $S_i$ by using its private key $K_i$ and uses it in its polynomial.

5.3.9. Exchange of shares and intermediate values between hosts is done by using Public key $y_i$ and group key $K_M$.

5.4. Generation of Random values and unique polynomial to be used in SMC

The CTA in a cloud $C_i$ chooses a set of random values ‘$R$’ from Galois Field $\mathbb{GF(t)}$ where $t$ is number of hosts in cloud in SMC such that $R = \{r_1, r_2, \ldots, r_n\}$ Where $r_1, r_2, \ldots, r_n \in \mathbb{Z}_t$ which is used in generating group key and publishes to hosts in a cloud. In an Ideal model if all the hosts are honest then the each host chooses an unique polynomial such that

\[ f(x) = \sum_{i=0}^{n-1} (a_i, x^i) \quad n \in \mathbb{N} \]  

\[ Q = \{q_0, q_1, q_2, \ldots, q_n\}, \text{ such that } Q \in \mathbb{Z}_p^* \text{ and } q_0 = p_i \]

The co-efficient of the term $x^n$ in the unique polynomial assumed by each host $H_i$ is the secret key $S_i$ sent by the CTA.
Therefore $f_i(0)=S_{ij}=\sum_{t=0}^{2n-2} (q_t x^t)$

The random values sent by the CTA to each host are $R=\{r_1, r_2, \ldots, r_n\}$ if $k=n-1$ and $R=\{r_1, r_2, \ldots, r_n, \ldots, r_{2n}\}$ if $k=2n-2$

5.5. Distribution of shares in a Cloud $C_i$ among the hosts

The shares are distributed among the hosts in cloud without HLTA. In the distribution phase, $n$ hosts decide on the degree ‘$K$’ of a polynomial as in the Shamir’s secret sharing

$$K = \begin{cases} 0 \text{ to } n-1, & \text{if all hosts are honest} \\ 0 \text{ to } 2n-2, & \text{otherwise} \end{cases}$$

Each host computes the share, $S_{hi}=f_i(r_i)$ and send share $S_{hi}$ to each $H_i$ with their random values $r_i$. The hosts distribute the shares by encrypting with the public key $y_i$ and decrypting with group key $K_M$ of the cloud. Since the shares obtained by each host are from different polynomials sum of secret keys cannot be revealed.

5.6. Intermediate results computation by each host in cloud

After each host $H_i$, receiving its shares it decrypts them using the same mechanism used to encrypt the shares. Then each host computes the Intermediate result in two cases,

**Case i:** $\text{IR}_i = \sum_{t=1}^{n} S_{ht}$, for $n$ random values. When $k=n-1$

**Case ii:** $\text{IR}_i = \sum_{t=1}^{2n} S_{ht}$, for $n$ random values. When $k=2n-2$

Then each host $H_i$ sends its $\text{IR}_i$ to the remaining hosts in an encrypted form. Each host $H_i$ receives the encrypted intermediate results and decrypts them to find the final sum polynomial. The $\text{IR}_i$ is also sent to the CTA by each host $H_i$ at this phase.

5.7. Generation of Final sum polynomial at each host:

CTA finds the sum polynomial from the shares/intermediate results received from the hosts $H_i$ which will be used for verification and identification of malicious host in the group. Each host agrees upon the final polynomial in the either of the cases.

**Case i:** if $k=n-1$ then the generalized sum polynomial is $G(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + a_{n-3}x^{n-3} + \ldots + a_0x^0$

The sum polynomial $G(x)$ is the sum of all the unique polynomial at each host $H_i$ that is $G(x)=\sum_{i=1}^{2n} f_i(x)$

**Case ii:** If $k=2n-2$ then, the generalized sum polynomial is $G(x)=a_{2n-2}x^{2n-2} + a_{2n-3}x^{2n-3} + \ldots + a_0x^0$

The sum polynomial $G(x)$ is the sum of all the unique polynomials at each host $H_i$ such that,

For each $j=1$ to $2n$ Compute $\sum_{t=1}^{2n} S_{ht} = \text{IR}_j$

The honest host can obtain the sum polynomial even though there might exists some malicious $\text{IR}_i$ sent by the malicious host. Each host $H_i$ substitutes each random value $r_i$ in the generalized sum polynomial $G(x)$ and the corresponding intermediate result $\text{IR}_i$ is equalized it received and each host obtains the following set of equations, in form $AX=b$
\begin{align*}
& a_{2n-2}r^{2n-2} + a_{2n-3}r^{2n-3} + a_{2n-4}r^{2n-4} + \cdots + a_0 r^0 = \text{IR}_2 \\
& a_{2n-2}r^{2n} + a_{2n-3}r^{2n} + a_{2n-4}r^{2n} + \cdots + a_0 r^{2n} = \text{IR}_{2n}
\end{align*}

From the above set of equations, an augmented matrix is generated to solve the final sum polynomial. The augmented matrix is, From the above matrix each host \( P_i \) chooses \((2n-1)\) rows with the corresponding \( \text{IR}_i \). The possible different ways to choose such alternatives are \( 2n \). The above augmented matrix is solved by to obtain the co-efficient of sum polynomial \( S(x) \) by \( n \) hosts and CTA.

5.8. Identifying malicious parties by CTA:

i) On obtaining the final polynomial \( S(x) \) at each host \( H_i \), \( r_i \) is substituted, For each \( j=1 \) to \( 2n \) Compute

\[
\sum_{i=0}^{2n-2} a_i r^i, \quad \ldots \ldots \ldots \ldots 1
\]

The outcome of the above step must be equal to the corresponding \( \text{IR}_i \). In step 1, if the above substitution does not equal to its corresponding \( \text{IR}_i \) of any \( H_i \) then the honest host comes to a conclusions that there exists a malicious host \( H_i \) and reports to CTA, as some \( H_i \) is malicious. As CTA also knows the final polynomial \( S(x) \), it in turn asks the Corresponding host \( H_i \) who might be malicious to resend the data or its data will be ignored. CTA finally sends the \( S(x) \) to the HLTA by encrypting with public key \( e \). Encryption Mechanism by CTA is given by \( C = M^e \mod n \).

ii) HLTA Verification:

On receiving \( S(x) \) from CTAs HLTA decrypts with keys \( d_{i1} \) and \( d_{i2} \) i.e., Compute \( M_i = C^{d_{i1}} \mod p_i \) for \( 1 \) to \( k \)

Apply the CRT to the \( M_i 's \) or each cloud to obtain \( M = C^{d_{i2}} \mod n \).

If \( 'M' \) it does not matches with the value required it finds that some CTA is malicious. If HLTA gets the same \( S(x) \) from all CTAs then all the hosts are honest and final output is accepted. HLTA finds out that the constant term in \( S(x) \) must be equal to the secret key sent by it to all \( H_i \) at the beginning of the Computation. At any case, if HLTA computes different \( S(x) \) from CTA then it concludes that CTA is colluded with malicious host or CTA itself is a malicious so its data will be ignored for final result or ask for repeat the computation.

6. CONCLUSION

In this paper we propose a secure multi host group privacy preserving scheme for cloud environment which the hosts in different clouds will share the data between them and with other clouds (which may be public or hybrid) securely under the control of CTA and HLTA. The shared secret key \( 'K' \) between the hosts is obtained from the credentials of the cloud, which is shared between \( n \) hosts during computation in encrypted form. The encryption and decryption is done by technique Carmichael-RPrime-RSA and Group key mechanism in cloud which provides authentication and data integrity. Here CTA and also HLTA detects and identifies malicious parties (almost \( n/2 \)) at two levels which provides more accuracy for final result. Since we are using Clouds for SMC it is also practically possible due to the characteristics of cloud. SMC will be provided as part of SaaS in cloud.

REFERENCES:


