AN EFFICIENT THRESHOLD CRYPTOGRAPHY SCHEME FOR CLOUD ERP DATA

Arnold Mashud Abukari¹, Edem Kwedzo Bankas² and Mohammed Iddrisu Muniru³

¹Computer Science Department, Tamale Technical University, Tamale, Ghana
²Computer Science Department, University for Development Studies, Navrongo, Ghana
³Mathematics Department, University for Development Studies, Navrongo, Ghana

ABSTRACT

Cloud ERP is a new concept that has caught the attention of researchers and industry players. One of the major challenges towards migrating ERP data to cloud is security and confidentiality. A number of secret sharing schemes (SSS) have been proposed Mignote, Asmuth-Bloom, Shamir and Blakley but not without challenges. In this research, we proposed an efficient threshold cryptography scheme to secure cloud ERP data. We modified the Asmuth-Bloom SSS and applied it to cloud ERP data.

KEYWORDS

Threshold Cryptography, Secret Sharing Scheme (SSS), CRT, Cloud ERP.

1. INTRODUCTION

Threshold Cryptography is a concept for sharing very sensitive data among a group of users. In our case, the group of users will be referred to as a group of n Cloud Enterprise Resource Planning (ERP) providers. Researchers over the years have succeeded in proposing Secret Sharing Schemes (SSS) including Asmuth-Bloom SSS [1] which is based on the Chinese Remainder Theorem (CRT). In [1] special sequence of integers and sequence of pairwise integers were used to share a randomly chosen secret to different users. The proposed scheme in [1] failed to handle data redundancy and not applicable to cloud. Shamir [2] also developed a SSS based on polynomial interpolation. In 1983, Mignotte [3] proposed a weighted SSS based on the CRT.

The scheme in [3] allowed the distribution of load among server (clouds) based on existing resources and the choice of a set of pairwise coprimes. In threshold cryptography, the most important requirement is the availability of a computable function without compelling parties to disclose their secret shares. This concept is known as function sharing problem and requires each part of the computation is carried out by different users without disclosing their secrets to each other. Several proposed schemes can be found in literature [[4],[5],[6],[7]] but most of them are built from Shamir [2] published in 1979. The concept of cloud computing has spread very fast and caught the attention of researchers and organisations due to the security implications on big data [8]. One of the new emerging solutions for cloud providers is Cloud ERP [9]. Cloud ERP, just like the on-premise ERP generates and stores highly sensitive data and thus requires maximum data protection from hackers and cloud providers. ERP data stored in cloud uses computational resources provided by the cloud provider and this raises questions on confidentiality, reliability and safety of the Cloud ERP data. In this research, we propose Threshold Cryptography (TC) scheme to help reduce redundancy and enhance the security of Cloud ERP data. To ensure confidentiality and integrity of cloud ERP data, our TC Scheme is applicable.
2. Threshold Cryptography

A cryptosystem is referred to as a threshold cryptosystem, if it requires several parties usually more than some threshold number in order to decrypt an encrypted message or data[6]. The security operations such as encryption, decryption, signature generation and verification can be performed by a group of processes without reconstructing the shared secrets.

Considering a general problem of \( t \)-out-of-\( n \) secret sharing in threshold cryptography, and we have \( n \)-parties to share \( x = (x_1; \ldots; x_n) \), two main important properties needs to be guaranteed. These properties are: Recoverability: Given any \( t \)-shares, we can recover \( x \) Secrecy: Given any \( \leq t \), \( x \) can never be recovered

3. Secret Sharing Scheme

Secret Sharing Scheme is applied to build a storage system using \( n \)-number of clouds independently and separately so that any of the clouds can recover their respective data without knowing the secret shares of the others. The shared data can only make meaning when a certain number of the cloud providers conspire. In this research, we examined the Asmuth-Bloom scheme [1], Shamir [2] and Mignotte’s scheme [3] to see how it can be applied to Cloud ERP Data. We propose Threshold Cryptography (TC) scheme to help reduce redundancy and enhance the security of Cloud ERP data.

3.1. Shamir Secret Sharing Scheme

The first scheme for sharing a secret was proposed by Shamir (Shamir, 1979) based on polynomial interpolation.

To obtain a \( (t, n) \) secret sharing, a random polynomial
\[
f(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_0
\]
is generated over \( \mathbb{Z}_p(x) \) where \( p \) is a prime number and \( a_0 = d \) is the secret.

The share of the \( i^{th} \) party is \( y_i = f(i), 1 \leq i \leq n \). If \( t \) or more parties come together, they can construct the polynomial by Lagrange interpolation and obtain the secret, but any smaller coalitions cannot.

3.2. Blakley Secret Sharing Scheme

Blakley gave a general construction of a threshold scheme using projective geometry.

Construction.

- i. In PG \((t, q)\), randomly select a hyperplane \( X \) to be the secret.
- ii. Choose \( n \) points in the hyperplane that are in general position in \( X \). Hence, any \( t \) points form a basis for \( X \).
- iii. To reconstruct the secret, we simple take the span \( U \) of the participating points and see if \( U = X \)
3.3. Asmuth-Bloom Scheme

The Asmuth-Bloom scheme uses a special sequence of integers and pairwise relatively prime integers $p_0[1]$. In this scheme, $p_1 < p_2 < \ldots < p_n$ are selected such that:

$$p_0 \prod_{i=0}^{k-2} p_{n-i} < \prod_{i=1}^{k} p_i$$

Where $k$ is the number of distinct shares. A secret $S$ is chosen as a random element of the set $\mathbb{Z}_{p_0}$.

Let the shares $I_i$ be chosen as $I_i=(S+.p_0) \mod p_i$, for $1 \leq i \leq n$ where $\lambda$ is an arbitrary integer or a positive integer generated randomly such that $S+.p_0 \in \mathbb{Z}_{p_0}; \ldots; \mathbb{Z}_{p_k}$. The Asmuth-Bloom scheme assumes four conditions as follows:

1) $p_0 > S$
2) $\forall i \in [1, \ldots, n], p_1 > p_0$
3) $p_1 < p_2 < \ldots < p_n$
4) $p_0 \prod_{i=0}^{k-2} p_{n-i} < \prod_{i=1}^{k} p_i$

The Asmuth-Bloom scheme requires $n$-times the computational resources for data storage for data like Cloud ERP data[1]. Given $k$ distinct shares $I_1, I_2, \ldots, I_k$, the secret $S$ can be obtained since $S=x_0 \mod p_0$, where $x_0$ can be calculated using the standard variant of the Chinese Remainder Theorem with a unique modulo $p_1; \ldots; p_k$ of the system.

$$x \equiv \left\{ \begin{array}{l}
I_{i_1} \mod p_{i_1} \\
\ldots \\
I_{i_k} \mod p_{i_k}
\end{array} \right.$$ 

To make the scheme become asymptotically ideal SSS and also to secure the confidentiality of cloud ERP data in case there is a cloud conspiracy, we introduce constraints and compactness to the Asmuth-Bloom Scheme such that $p_0 < p_1 < p_2; \ldots; p_n < 2p_0$.

4. Problem Statement

The concept of Threshold Cryptography in the application to cloud computing has not received adequate research after Shamir, Blakley, Asmuth-Bloom and Mignotte made strides in applying it to secret sharing. The desire of organisations for centralised, accurate and timely information that will help in analysing data in order to make strong strategic decisions and gain a favourable competitive advantage has given birth to Cloud Enterprise Resource Planning (ERP). The size of encrypted ERP Data in cloud, the execution time and memory usage during encryption and decryption deserves much attention and improvement and has been the focus of our research.
5. **PROPOSED SCHEME**

In this research we propose a new Scheme based on CRT to help reduce the execution time for encryption and decryption as well as memory usage. Our scheme also reduces the size of encrypted data. We propose a Scheme to provide security for cloud ERP data and also to significantly reduce the redundancy of Cloud ERP Data.

Considering our modification on the four conditions that needs to be satisfied to fulfill the Asmuth-Bloom secret sharing scheme as stated in (Asmuth and Bloom, 1983), we propose a new Scheme to provide security for cloud ERP data and also to significantly reduce the redundancy of Cloud ERP Data when the Mignotte scheme (Mignotte, 1983) is applied. The proposed scheme also applied the Asmuth-Bloom scheme to prove its threshold cryptographic properties. We considered the following notations in building the our Scheme: D – Cloud ERP Data, p1; p2; p3; : : : ; pn being pairwise coprime numbers, p0 is an arbitrary integer also called the PROPOSED Key that corresponds with the above coprime numbers.

We considered the following notations in building the Scheme:

- D – Cloud ERP Data
- p1, p2, p3, …, pn being pairwise coprime numbers, p0 is an arbitrary integer also called the PROPOSED Key that corresponds with the above coprime numbers.
- We propose that the Cloud ERP data be stored among n number of cloud service providers.

We propose that

$$\prod_{i=1}^{t} p_i > p_0$$

Where p0 is the PROPOSED key, t is the threshold for the cloud providers in order to recover the secret or ERP data.

To calculate the ERP data shares for each cloud service provider we use:

$$C = (D + \gamma \cdot p_0) \mod p_i; \forall i \in [1, ..., n]$$

C is the cloud service providers shares, D is the data or secret, p0 is the PROPOSED key and \(\gamma\) is an arbitrary integer randomly generated.

5.1. **Conditions For Our Scheme**

The conditions that needs to be satisfied by our scheme are:

- \(p_0 > D\)
- \(\prod_{i=1}^{t} p_i > p_0\)
- \(p_0 = p_1 < p_2 < \cdots < p_n < 2p_0\) (Constraint compactness)
- \(\prod_{i=0}^{n-2} p_{n-1} < \prod_{i=1}^{t} p_i\)

The condition four(4) is necessary to confirm the threshold property of our scheme needed for secret sharing schemes.
5.2. Proposed Algorithms

**Algorithm 1** OUR SCHEME SHARING PHASE ALGORITHM

To share an ERP Data $D$ among a group of $n$ cloud service providers, we compute the following:

1. Select a set of pairwise relatively prime integers $p_0 < p_1 < \ldots < p_n$ where $p_0 > d$ is a prime such that

   \[ p_i \prod_{i=1}^{t} > \prod_{i=0}^{t-2} p_{n-1} \]

2. Let $M$ denote $\prod_{i=1}^{t}$

3. We compute $y = d + \alpha m_0$ where $\alpha$ is a positive integer generated randomly subject to the condition that $0 \leq y < M$

4. The share of the $i$th cloud user where $1 \leq i \leq n$ is

   \[ y_i = y \mod p_i \]

**Algorithm 2** OUR SCHEME CONSTRUCTION PHASE ALGORITHM

Assume $S$ is a coalition of $t$ cloud users needed to come together to construct the ERP Data. Let $M_s$ denote $\prod_{i \in S} p_i$

1. Given the system $y \equiv y_i \mod p_i$ for $i \in S$, we solve $y$ in $\mathbb{Z}_{p_0}$ using the Chinese Remainder Theorem (CRT).

2. Compute the ERP Data as

   \[ D = y \mod p_0 \]

3. Using CRT, $y$ is determined uniquely in $\mathbb{Z}_{M_s}$ since $y < M_s \leq M$. The solution will also be unique in $\mathbb{Z}_M$

**Algorithm 3** PROVE OF OUR SECRET SHARING SCHEME ALGORITHM AS BEING PERFECT

1. Assume $S^1$ is a coalition of size $t - 1$ cloud users has gathered.

2. Let $y^1$ denote the unique solution for $y$ in $\mathbb{Z}_{M_s}$

3. Consider $\prod_{i=1}^{t} > p_0 \prod_{i=1}^{t-1} p_{n-i+1}$

4. We have $M/M_{S^1} > p_0$, hence $y^1 + jM_{S^1}$ is smaller than $M$ for $j < p_0$ and there are $p_0$ of them. The $D$ can be any integer from $\mathbb{Z}_{p_0}$ and the coalition of $S^1$ has no information on $D$ and hence our scheme is a perfect secret sharing Scheme.
5.3. Cloud Erp Data Redundancy With Our Scheme

Considering $p_o$ being the PROPOSED key, $p_i$ being the ERP data shares based on coprime randomly selected numbers and $D$ is the ERP data then the redundancy of Cloud ERP data based on our proposed scheme is defined as follows per (Mignotte, 1983):

$$\left[\sum_{i=1}^{n} \log_2 p_i \right] / \left[ \log_2 p_0 \right] \leq \left[\sum_{i=1}^{n} \log_2 p_i \right] / \left[ \log_2 D \right]$$

Where $n$ is the number of cloud service providers.

6. RESULTS AND DISCUSSION

We used a pdf file of size 14,752,894 bytes (14mb) and the number of cloud service providers $n$ to be 5 whiles the $k$-value is 3. These common parameters were used for all three Secret Sharing Schemes namely Shamir SSS, Blakley SSS and our Scheme.

| File type | File Size | Operation type | $n$ | $k$ |
|-----------|-----------|----------------|-----|-----|
| pdf       | 14mb (14,752,894 bytes) | Encryption/Decryption | 5   | 3   |

Figure 1: Parameters for testing

6.1. Encryption Execution Time Analysis

Despite no differences in memory usage compared with Blakley and Shamir, our scheme uses lesser time during the encryption process and thus per the simulation results is faster during encryption compared to the Shamir, Blakley and Asmuth-Bloom Schemes. Faster execution time means less computational resources are used during the encryption process. This lesser time for encryption will compensate for the high memory usage since the memory will be released within the shortest time frame compared to others.
6.2. Encryption Memory Usage Analysis

Our scheme performed equally in the memory used in the encryption process as Blakley and Shamir but performed better than Asmuth-Bloom scheme. The simulation results averagely suggests our Scheme is performing well in memory space just as the other schemes.

6.3. Size On Disk After Encryption Analysis

Encrypted Data size after the encryption process was also investigated by comparing all the three schemes. The simulation results averagely suggests that our scheme performs better in terms of the size of encrypted data. Comparing the sizes of the encrypted data from the schemes reveals the size of the our encrypted data is relatively smaller compared to others. This suggests that our scheme improves the efficient use of space on the servers.
6.4. Decryption Execution Time

In the decryption process, Our scheme performs better in memory usage and time of execution. This suggests that our scheme saves time and computational resources during the decryption processes.
7. Conclusion

In this research work, we propose a new secret sharing scheme which is based on the Chinese Remainder Theorem (CRT). The main objective behind our scheme is to help avoid cloud collusion using threshold cryptography and to reduce the execution time and memory usage as well reduce the size of encrypted data in cloud. The PROPOSED scheme is an improvement to the Shamir, Blakley and Mignotte schemes since it reduces the exection time and computational resources. The test runs results show an improved execution time and less computational resources when our scheme is applied.

References

[1] C. Asmuth and J. Bloom, “A modular approach to key safeguarding,” IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 208–210, March 1983.

[2] A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, pp. 612–613, Nov. 1979. [Online]. Available: http://doi.acm.org/10.1145/359168.359176

[3] M. Mignotte, “How to share a secret,” in Cryptography, T. Beth, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 371–375.

[4] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, “How to share a function securely,” in Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing, ser. STOC ’94. New York, NY, USA: ACM, 1994, pp. 522–533. [Online]. Available: http://doi.acm.org/10.1145/195058.195405

[5] H.-F. Huang and C.-C.Chang, “A novel efficient (t, n) threshold proxy signature scheme,” Information sciences, vol. 176, no. 10, pp. 1338–1349, 2006.

[6] Y. Desmedt, “Some recent research aspects of threshold cryptography,” in International Workshop on Information Security. Springer, 1997, pp. 158–173.

[7] V. Shoup, “Practical threshold signatures,” in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2000, pp. 207–220.

[8] C. P. Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on big data,” Information Sciences, vol. 275, pp. 314–347, 2014.

[9] C. Songsheng and Y. Peipei, “Economic benefits of enterprise resources planning (erp)-based on empirical evidence from chinese listed companies,” in Logistics Systems and Intelligent Management, 2010 International Conference on, vol. 3. IEEE, 2010, pp. 1305–1308.