Asian Journal of Information Technology 14 (2): 74-83, 2015
ISSN: 1682-3915

Analysis and Comparison the Security and Performance of Secret Sharing Schemes

Aisha Abdallah and Mazleena Salleh
Universiti Teknologi Malaysia, Johor, Malaysia

Abstract: Distributed storage systems merge a number of storage sites into a collective whole. Files are split into smaller pieces which are computationally manipulated and then distributed to the storage sites. It recovers some subset of pieces when a client wants to read a file whereas those pieces are combined to reconstruct the original file. Distributed storage systems are better than traditional single-site storage systems because they offer a variety of benefits such as availability, proximity and reliability. One of the basic methods used to disperse the data in storage systems are secret sharing schemes. In cryptography, secret sharing is a technique to share a secret among a group of members, each of which holds a portion of the secret. The secret can only be retrieved when a certain number of members combine their shares together while any combination with fewer shares has no extra information about the secret. There are many secret sharing schemes and each one achieves a different level of security with different performance and storage requirements. In this study, we analyze the security and performance of three secret schemes which are the most common schemes for information dispersal used within distributed storage systems, namely Shamir's scheme, Rabin's IDA and hybrid scheme. Several tests were conducted to understand the fundamental concept of the schemes as well as to explore the security, performance and the capability of the schemes.

Key words: Shamir's secret sharing scheme, Rabin's IDA, Krawczyk's hybrid sharing scheme, sites, client

INTRODUCTION

Securing data in distributed storage systems are considered one of the biggest challenges in information security system. Some of these challenges include: the data in those systems are threatened by damage, loss or leakage; unauthorized access to those systems is considered the biggest threat; users must be assured that query they send to the distributed storage system as well as returned data (query result) from the distributed storage system is not tampered with by adversaries; the data can be modified by unauthorized users. Cryptography as a solution to above common problems in the distributed storage system is used to verify the security services Confidentiality, Integrity and Availability (CIA) (Nirmala *et al*., 2012). There are cryptographic techniques used in the distributed storage system such as Data encryption, Homomorphic Encryption (Gentry, 2009), Secret Sharing algorithms and Private Information Retrieval (PIR) (Chor *et al*., 1998). However, while PIR and Homomorphic Encryption can ensure the confidentiality of data, they have computational costs. In addition adversaries can affect both throughput and latency (Kantarcioglu and Clifton, 2005). Furthermore, Data encryption is insufficient to ensure the security of data because it is still threatened by lose, theft or damage that making it unavailable. Secret sharing schemes provide both data availability and a certain degree of data confidentiality with low computational and storage costs compared with other cryptography techniques. Thus, this study will focus mainly on these schemes and their security and performance. Particular focus will be on Shamir's scheme, Rabin's IDA and Krawczyk's hybrid scheme which is called (Secret Sharing Made Short). Which are commonly used in distributed storage systems (Slamanig and Hanser, 2012). A secret sharing scheme is an important tool to protect distributed file systems against data damage, loss, destruction and leakage. The basic idea of secret sharing introduced by Shamir and Blakley independently (Shamir, 1979; Blakley, 1979) is that an administrator disperses a piece of information (called a share) about the secret to each participant such that a subgroup of participants have privileges to recover the secret but unprivileged subgroup of participants cannot obtain any information about the secret. Secret sharing schemes are now under consideration in distributed storage systems, particularly for cloud systems because they can distribute data to multiple servers and are resistant to system failures caused by natural disasters or human error (Takahashi *et al*., 2013). Ermakova and Fabian (2013) and Fabian *et al*. (2014) use secret sharing for health data in multi-provider clouds. Furthermore, secret sharing schemes are vital tools in visual cryptography (Annamalai and Thanushkodi, 2013). There are many types of secret sharing schemes as shown in Fig. 1. This

Corresponding Author: Aisha Abdallah, Universiti Teknologi Malaysia, Johor, Malaysia

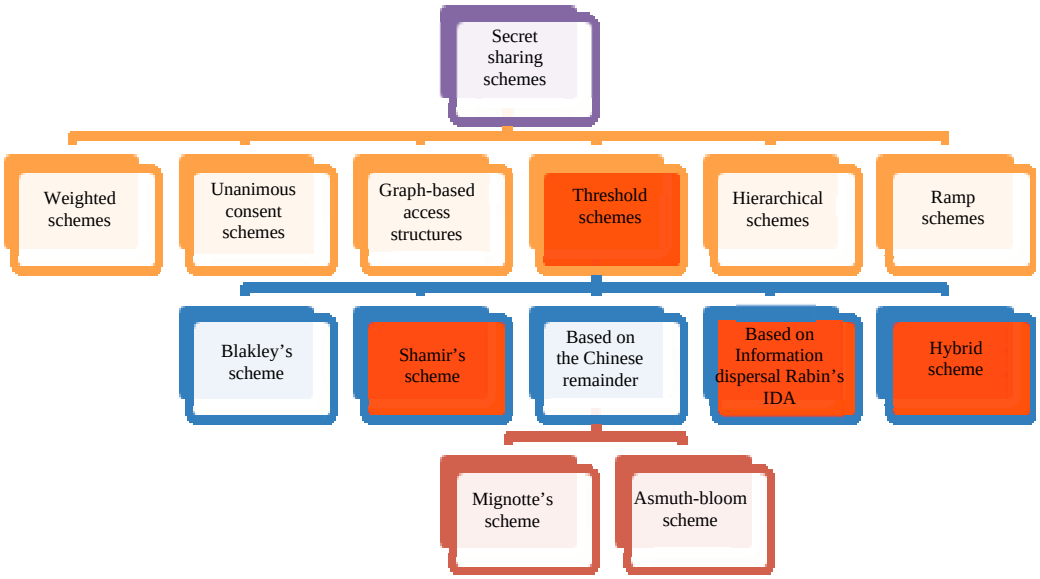

Fig. 1: Types of secret sharing scheme

study is going to focus only on threshold secret sharing schemes. The following sections will present definition and explanation for threshold secret sharing schemes. Before that significant notations and symbols should be identified:

- n number of shares labeled with the numbers 1, ..., n
- Each share will be distributed into nodes of storage system
- m is the threshold value
- F or S will indicate the secret
- F1, ..., Fn or S1, ..., Sn will indicate the shares of F or S, respectively
- A dealer (administrator) who is coordinating the secret sharing scheme
- In each secret sharing scheme there are two phases

Split the secret: The secret is constructed by dealer in some pre-defineddomain to derive the shares and then distribute the shares to the nodes.

Recover the secret: The nodes will reconstruct the secret after they contribute their shares.

MATERIALS AND METHODS

Threshold secret sharing schemes: A set of m shares (where m is the minimum share threshold) has the ability to rebuild the secret while set with threshold lower than m does not. This scheme is called a (m, n)-threshold scheme (Guo and Chang, 2013; Iftene, 2006). There are many types of threshold schemes as illustrated in Fig. 1. In this study, three secret schemes of securing data are analyzed namely Shamir's scheme, Rabin's IDA and hybrid scheme. Several tests were conducted to explore and investigate security and performance capabilities of these schemes such as strength of shares test, recover the secret of m shares and verify the integrity of the shares test were conducted from the security aspect. From the performance aspect, the flexibility of schemes test, the space efficiency test and time of processing test were conducted.

Shamir's secret sharing scheme: Shamir's (m, n) threshold scheme (Shamir, 1979), uses Lagrange interpolation with given m different points (x_i, y_i) of the form $(x_i, f(x_i))$ where f(x) is a polynomial of degree less than m where f(x) is decided by Eq. 1:

$$f(x) = \sum_{i=1}^{m} y_i \prod_{\substack{1\le j\le t \\ i\neq j}} \frac{x-x_j}{x_i-x_j} \quad (1)$$

Shamir's scheme is defined for a secret $s \in Z/pZ$ with p prime by setting $a_0 = s$ and choosing $a_1, a_2 \ldots a_{m-1}$ at random in Z/pZ. The trusted party computes f(i), for all $1 \le i \le n$ where:

$$f(x) = \sum_{k=0}^{m-1} a_k x^k \quad (2)$$

The shares (i, f(i)) are distributed to the n different parties. Since, the secret is the constant term $s = a0 = f(0)$, the secret is reconstructed from any m shares (i, f(i)) for i belong to {1, ..., n} by:

$$s = \sum_{i\in I} c_i f(i) \text{ where } c_i = \prod_{\substack{j\in I \\ j\neq i}} \frac{i}{i-j} \quad (3)$$

Rabin's IDA: Rabin's Information Dispersal Algorithm (IDA) splits a file F of length L = |F| into n pieces Fi, $1 \le i \le n$, each of length |Fi| = L/m, so that every m piece suffices for recovering F. Dispersal and reconstruction are computationally efficient. The sum of the lengths |Fi| is (n/m)×L. Since, n/m can be chosen to be close to 1, the IDA is space efficient (Rabin, 1989). In practice, Rabin's IDA is implemented based on the explained procedure. First, the original file F is divided into m segments S1, S2, ..., Sm, each of size L/m. Then, the m segments are encoded into n unrecognizable pieces F1, F2, ..., Fn using a m of n erasure code such as Reed Solomon code which used Vandermode matrix as coded or dispersed matrix (Plank and Ding, 2005). The practical implementation of the Rabin's IDA is represented as a matrix-vector product, a generator or dispersal matrix G is created which has n rows and m columns. This matrix is multiplied by an m-element vector D (called the data or message) to yield a n-element vector C called the codeword. Each element of the codeword is stored in a different storage node:

$$G_{n\times m} \times D_{m\times 1} = C_{n\times 1}$$

$$\begin{bmatrix} g_{1,1} & g_{1,2} & L & g_{1,m} \\ g_{2,1} & g_{2,2} & L & g_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ g_{n,1} & g_{n,2} & L & g_{n,m} \end{bmatrix} \times \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad (4)$$

Then, the matrixes:

$$A = \begin{bmatrix} I \\ G \end{bmatrix}$$

And:

$$E = \begin{bmatrix} D \\ C \end{bmatrix}$$

will be configured where I is identity matrix such that matrix A and E adhere to the following equation such that each element of the matrix E is stored on a different storage node (Resch and Plank, 2011):

$$A \times D = E$$

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ g_{1,1} & g_{1,2} & g_{1,3} & \cdots & g_{1,m} \\ g_{2,1} & g_{2,2} & g_{2,3} & \cdots & g_{2,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ g_{n,1} & g_{n,2} & g_{n,3} & \cdots & g_{n,m} \end{bmatrix} \times \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad (5)$$

To recover the data D from m dispersed pieces corresponding to n rows of E, the sub-matrixes A' and E' must be obtained from A and E matrixes such that each node in the system has a corresponding row of the matrix A and the vector E. When a node fails, the failure is reflected by deleting the node's row from A and from E and so the sub matrixes A' and E' are obtained as described in Eq. 6:

$$D = A'^{-1} \times E \quad (6)$$

A generator or dispersal matrix G is Vandermonde matrix in which every n row is linearly independent. Entry of a Vandermonde matrix that adheres to this equation is:

$$g_{i,j} = i^{j-1} \quad G = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & n \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 2^{m-1} & \cdots & n^{m-1} \end{bmatrix} \quad (7)$$

Each entry in matrix D is an integer which entering into computer words, it is convenient that w be a power of two. To achieve this Galois Field arithmetic, GF (2^w) employed where the addition n is equal to bitwise exclusive-or (XOR) and multiplication is implemented in a variety of ways either in hardware or software (Annamalai and Thanushkodi, 2013).

Hybrid secret sharing scheme: Hybrid secret sharing scheme was proposed by Krawczyk (1994). It is a blending of Rabin's idea of an Information Dispersal Algorithm (IDA) and Shamir scheme in order to reduce the size of user's shares with ensuring the integrity and availability of shares thus Krawczyk's hybrid scheme is called Secret Sharing Made Short. The hybrid scheme components are as follows:

- A symmetric encryption scheme with k-bit keys
- Shamir secret sharing scheme
- Rabin Information Dispersal algorithm
- One way hash function
- Error correction code

An Error-Correcting Code (ECC) or Forward Error Correction (FEC) code is a system of adding redundant data or parity data to a message such that it can be recovered by a receiver even when a number of errors were introduced, either during the process of transmission or on storage. Since, the receiver does not have to ask the sender for retransmission of the data, a back-channel is not required in forward error correction and it is therefore suitable for simplex communication such as broadcasting. There are two types of error-correcting codes which are

bit by bit basis codes and block-by-block basis codes such as Reed-Solomon (RS) codes (Saramentovas and Ruzgys, 2007).

Algorithm 1 (Pseudo code for hybrid secret sharing scheme):

Split secret (M) procedure
K←Select random key
C←Encrypt K(M)
$C_1, C_2,...,C_n$←IDA (C)
$K_1, K_2,...,K_n$←Shamir(K)
For i←1 TO n DO
H[i]←Hash (K[i] C[i])
Ri←ECC (H[i])
FOR i←1 TO n DO
S[i]←K[i]C[i] R1[i]···Rn[i]
RETURN S[i]
Recover secret M with threshold t procedure
FOR i←1 TO n DO
K[i]C[i] R1[i]···Rn[i]←S[i]
FOR i←1 TO n DO
H[i]←Recover ECC (Ri, t)
FOR i←1 TO n DO
Check IF Hash (K[i] C[i]) = H[i]
THEN C←Recover IDA ($C_1, C_2,...,C_t$)
K←Recover Shamir ($K_1, K_2,...,K_t$)
M←Decrypt K(C)
RETURN M
Otherwise return erorr

Split secret (M) procedure: The constructor node of the secret is responsible for creating the secret and distributing it into shares over nodes; first it selects a random key to encrypt the data file and then this node will use Rabin's IDA to split the encrypted data file into C_1, C_2, ..., C_n and Shamir's scheme to split the used key into K_1, K_2, ..., K_n in order to distribute them over n nodes. However, before that this node will attach R1[i] ,..., Rn[i] with each split concatenated pair (C_i, K_i). R1[i], ..., Rn[i] is error correction code for hashing concatenate of C_i with K_i. Hence, the components of each share are a split piece of cipher text and a key with coded hash (Guo and Chang, 2013; Rogaway and Bellare, 2007; Benaloh and Leichter, 1993).

Recover secret M with threshold t procedure: The reconstructor node will parse each share of any n nodes into its components strings K[i], C[i], R1[i], ..., Rn[i] then from n Ri the decoded error correct code will compute to reconstruct t of hash values. The reconstructor node will then use the hash values to check the integrity of the parsed pieces K[i], C[i] to determine which returned pieces to be valid if there are invalid pieces then will be discarded and checked another new pieces. Otherwise, the known valid t pieces will be used by Rabin's IDA to recover the cipher text and Shamir's scheme to recover the encryption key. The recovered key will then be used to decrypt the cipher text and thus the message (secret) is recovered (Guo and Chang, 2013; Rogaway and Bellare, 2007).

Experimental setup: Five tests are being used to analyze the security and performance of the three schemes which are as follows:

Secutity tests:

- Test 1: to test the condition of each scheme that is retrieval of information with $m \leq n$
- Test 2: to test the ability to recover the secret compromising the integrity of any share of threshold shares by changing one, two or three bits of the any share, either to number or letter

Performance tests:

- Test 3: this test checks the flexibility of scheme in recovering the secret by entering shares in different order
- Test 4: storage requirements test checks the storage consumed for each share by testing the length of scheme's shares
- Test 5: there are three tests to analyze scheme's time for splitting and recovering which are
  - Test the effects of shares number (n) on the split and recovery time by testing varied number of shares
  - Test the effects of threshold value on the split and recovery time by testing varied values of threshold
  - Test the effects of data file size on the split and recovery time by testing different sizes of data file

RESULTS AND DISCUSSION

Security analysis of Shamir's scheme

Test 1 (retrieval of information with $m \leq n$): This experiment tested the condition of Shamir's secret sharing which is recovering of m shares to get the secret yet any m-1 shares provide no information about the secret. As the size of each share is equal to size of secret, this scheme has computational security.

Test 2 (check integrity of Shamir's scheme): Testing the ability to recover the secret compromising the integrity of any share of threshold shares by changing one, two or three bits of any share either to number or letter. Such that in all cases the Shamir's scheme returned error.

Performance analysis of Shamir's scheme: The tests contacted to analyze the performance Shamir's scheme are as follows:

Test 3 (flexibility analysis of Shamir's scheme): By combining shares in different order the secret are recovered.

Test 4 (storage requirement analysis for Shamir's scheme): Shamir's total storage requirements for data file of size |F| bytes and if the total number of shares is n then the total size is n×|F| bytes. This means the storage requirement for each share (piece) is |F| bytes. Therefore, the Shamir's scheme needs a high storage requirement whereas the size of each share stored in each node should be at least equal to the secret size (data size |F|). For this reason, this scheme is no longer used for sharing secret of data file; instead it is utilized for sharing encryption key as a secret because the encryption key has small size compared with data file (Wang *et al*., 2010). The following example explained the storage requirement of Shamir's scheme for data file. Suppose 4 kb is the size of data file or secret (F), the total shares n are 16 and the reconstruction threshold of 10 shares (m). There will be 16 shares (S) whereby S = $\{S_0, S_1, ..., S_{15}\}$. To apply the Shamir's scheme, consider F as 4096 individual bytes whereby F = $\{f_0, f_1, ..., f_{4095}\}$ and S_n will also consist of 4096 individual bytes of $\{s_{i,0}, ..., s_{i,4095}\}$ where i is the share number. As the storage requirement for each share is 4 kb (4096 bytes), the total storage requirement is n×|F| = 16×4 = 64 kb. Applying Test 4 demonstrates the length of each share of Shamir's scheme is the same as length of the secret.

Test (5a) (the effects of number of shares on the split and recovery time): Figure 2 shows the time of Shamir's scheme for splitting and recovering data file. It can be seen that the time increases with increasing in the number of shares n as it is obvious that the time taken to split 450 kb into 10 shares is less than the time taken to split into 40 shares.

Test (5b) (check the effects of threshold value on the split and recovery time): Shamir's scheme time for splitting data file is not impacted by varied values of threshold m. This is because the size of share in Shamir's scheme is equal to the size of data file |F| = 450 kb (secret) for any value m. The recovery time is increased by large values of m which means that the recovery time is influenced by threshold values. This test is illustrated in Fig. 3.

Test (5c) (check the effects of data file size on the spilt and recovery time): As shown in Fig. 4, the time taken by Shamir's scheme to split and recover the data file increases with increasing the size of data file.

Rabin's IDA security analysis: There are two tests conducted to analyze the security of Rabin's IDA which are as follows:

Test 1 (retrieval of information with n<m): This experiment tests the propriety of Rabin's IDA which is recovering of m shares to get the secret. Actually Rabin's IDA is the same Shamir scheme which means when shares are equal to predefined threshold m, the secret value can be returned. However, if the thresholds are fewer than

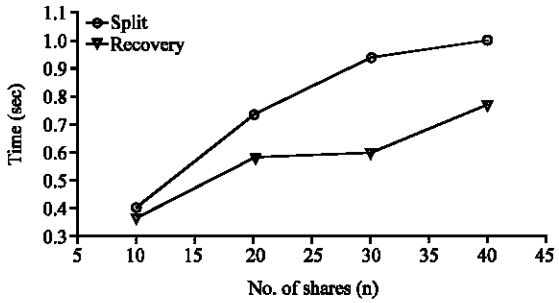

Fig. 2: Shamir's scheme time for splitting and recovering data file with varied n

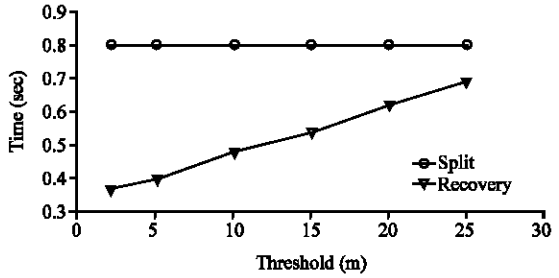

Fig. 3: Shamir's scheme time for splitting and recovering data file with varied m

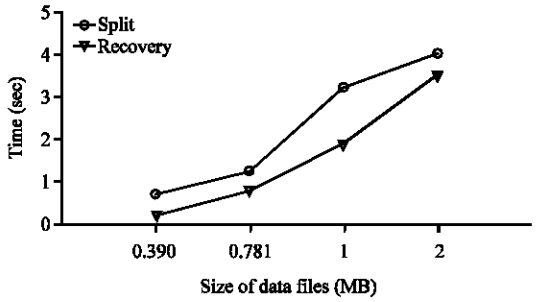

Fig. 4: Shamir's scheme splits time with varied sizes |F|

already predefined threshold m, the Rabin's IDA returned an error. Security of Rabin's IDA is weak because it has no randomness, unlike Shamir's scheme. If the generator matrix is known and the data have recognizable patterns (i.e., it is not random looking) then it is possible to guess the content of missing share. If one has m-1 shares, trying each of the 2^w possibilities for words of a missing share will yield m recognizable words when the correct value is attempted (Annamalai and Thanushkodi, 2013; Dautrich and Ravishankar, 2012).

Test 2 (check the integrity of Rabin IDA): Testing the ability to recover the secret compromising the integrity of any share of threshold shares by changing one, two or three bits of the any share, either to number or letter such that in all cases the Rabin's IDA returned error.

Rabin's IDA performance analysis: The tests conducted to analyze the performance Rabin's IDA are as follows:

Test 3 (flexibility analysis of Rabin's IDA): The result of flexibility test of Rabin's IDA is similar to Shamir scheme which by entering the shares in different order the secret will be recovered.

Test 4 (storage requirement analysis of Rabin's IDA): The Rabin's total storage requirement for data size |F| bytes with total number of shares n is n×(|F|/m) bytes where m is threshold, so the storage requirement for each share (piece) is (|F|/m) bytes. For example, suppose 4 kb is the data file size or (secret) that needs to be split into 16 shares and reconstructs this file by possessing some 10 shares thus m = 10 is a threshold. Whenapply Rabin' IDA, it will pad F to be 4100 bytes and then partition it into ten data shares F_{S0}, ..., F_{S9} of 410 bytes each. Each data share F_{si} is considered to consist of 410 individual bytes $F_{si, 0}$, ..., $F_{Si, 409}$. The storage requirement for each shareis 0.4003 kb and the total storage requirement is 16×0.4003 = 6.41 kb. The result of this test indicates that the size or length of each share is smaller than length of the secret.

Test (5a) (check the effects of number of shares on the split and recovery time): Figure 5 shows the time of Rabin's IDA for splitting and recovering data file where this time is increasing with increasing the number of shares n as it is obvious that the time taken to split 450 kb into 10 shares is less than the time taken for splitting into 40 shares. Similarly, in recovery time, recovery for big values of n such as 30 and 40 will be more than the time taken by small values of n.

Test (5b) (check the influences threshold value on the spilt and recover): Figure 6 shows the time in seconds taken to split/recover a 450 kb into/from 30 shares for different threshold values. Whereas the split time which is taken by big threshold values is less than the time taken by small threshold values because the size of split data (shares) has small size for big threshold values (Nirmala *et al*., 2012) while the recovery time is decreasedby small values of threshold and then it starts increasing with big threshold values.

Test (5c) (check the effects of data file size on the spilt and recover time): Figure 7 showed the time taken by Rabin's IDA for splitting and recovering data file. It is observed that there is considerable increase in split time as the size of data file is increasing. Recovery time, like the split time is also increasing with increases in size of data file. The whole consumed time for splitting these different sizes of the data file is more than the taken time for recovering them.

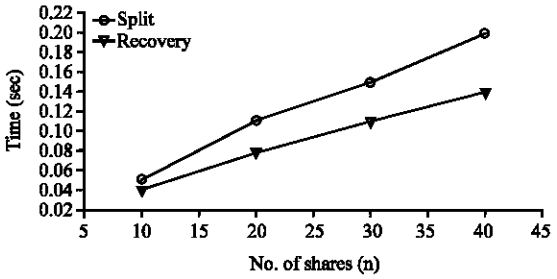

Fig. 5: Rabin's IDA time for splitting and recovering data file with varied n

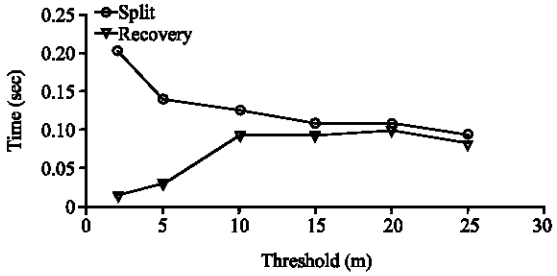

Fig. 6: Rabin's IDA splitting and recovering time with varied m

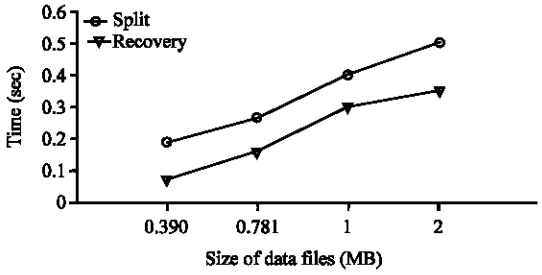

Fig. 7: Rabin's IDA split and recovery time with varied size of data files

Hybrid secret sharing scheme performance analysis: The performance analysis for hybrid scheme addresses the storage requirement and time taken by a hybrid scheme to split and recover a secret whereas the components of this scheme will be considered in this analysis.

Storage requirement analysis for hybrid scheme: The storage requirement for hybrid scheme is dependent on storage required by its components which are Encryption scheme with key, Shamir's scheme, Rabin's IDA, Hash function and Error correction code. Therefore, the storage requirement is storage consumed by Rabin's IDA to split a data file of size |F| into n nodes with threshold m is n×(|F|/m) that are added to storage required by Shamir's scheme to split encryption key of size |K| into n nodes is n×|K|; therefore, the hybrid scheme storage requirement n×(|F|/m+|K|) where the size of each share is (|F|/m+|K|) (Annamalai and Thanushkodi, 2013). It is observed that the storage consumed by a hash function and error correct code is ignored for a hash function is a compressed function and as a result the error correcting code for hashed data will also be small, therefore, the storage requirement for these two components are considered as negligible (Krawczyk, 1993). The same example used above for explaining the storage requirement for Shamir's and Rabin's schemes will be used here to explainthe storage requirement for the hybrid scheme. The data file (secret) size is 4 kb which is required to be shared into 16 nodes, meaning n = 16 and reconstructs this file by possessing some 10 shares thus m = 10 is a threshold. When using the hybrid scheme, a random 16 byte encryption key should be selected and the data should be encrypted with an encryption algorithm such as AES then split using Rabin's IDA and split the key using Shamir. The total storage requirement is 16×(410+16) = 6.65 kb.

Time analysis of hybrid scheme: The time of the hybrid secret sharing scheme is the same as the storage requirement. Both are influences by components of hybrid scheme performance that means the time for splitting and recovering the secret by hybrid scheme is dependent on the time required by its components which are the encryption scheme with key, Shamir's scheme, Rabin's IDA, hash function and error correction code. Therefore, the time of the hybrid scheme for splitting and recovering is calculated by Eq. 8. Table 1 explained each parameter in the equation:

$$TH_{\substack{split\\recover}} = T_{\substack{Enc\\Dec}} + TR_{\substack{split\\recover}} + TS_{\substack{split\\recover}} + T_{Hash} + TECC_{\substack{encoding\\decoding}} \quad (8)$$

The time tests in Table 1 are conducted to analyze hybrid scheme time, considering the following assumptions for this scheme:

- Encryption function ASE-128
- Length of encryption key 128
- MD4 hash function
- Classic reed solomon error correcting code

Test (5a) (check the effect of number of shares on the split and recovery time): As observed in Fig. 8, the time consumed by hybrid scheme for splitting and recovering data file is increased for large values of n.

Test (5b) (check the effects of threshold value on the spilt and recovery time): The time is decreased for large value of m which is why the size of split and recovered share is decreased for large value of m as illustrated in Fig. 9.

Test (5c) (check the influence of data file size on the spilt and recover time): As shown in Fig. 10, the time taken by Hybrid scheme for splitting and recovering data file is increasing with increasing the size of data file.

Schemas comparison: The above study presented and discussed the results of experiments which are conducted to analyze the security and performance characteristics of Shamir, Rabin and hybrid schemes. In this study, these results will be used for comparing among schemes. In the security aspect, the hybrid scheme has computational

Table 1: Time analysis of hybrid scheme

Parameters	Explanation
$T_{Enc/Dec}$	Time taken by encryption scheme which uses to encrypt and decrypt data file where the time taken is dependent on the choice of encryption function, for example time taken by AES-256 is more than the time taken by RC4-128 encryption function (Venkataramana and Padmavathamma, 2012)
$TR_{split/recover}$	Time taken by Rabin' IDA to split and recover the data file
$TS_{split/recover}$	Time taken by Shamir's scheme to split and recover encryption key
T_{Hash}	Time taken by hash function that computes on each share to ensure the integrity of a secret where the time taken is based on the choice of hash function, for example time taken by SHA-256 is more than the time taken by MD5 or MD4 hash functions (Venkataramana and Padmavathamma, 2012)
$TECC_{encoding/decoding}$	Time taken by error correcting code for encoding and decoding the hashed share in order to recover it in case of any error or modify on the share. Time of error correcting code in the hybrid scheme is the same as time of encryption and hash functions because it also based on choice of error correcting code algorithm such that the time taken by Classic Reed-Solomon codes is more than Cauchy Reed-Solomon codes (Pang *et al*., 2006)

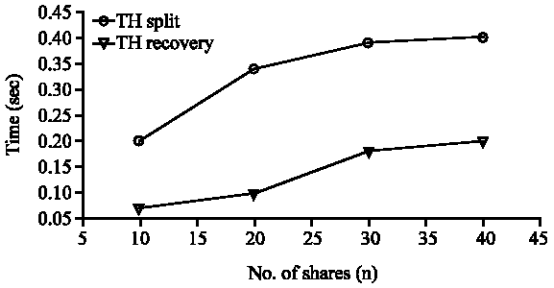

Fig. 8: Hybrid scheme time for splitting and recovering data file with varied number of shares (n)

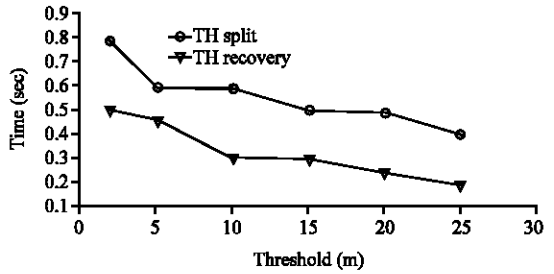

Fig. 9: Hybrid scheme time for splitting and recovering with varied threshold (m)

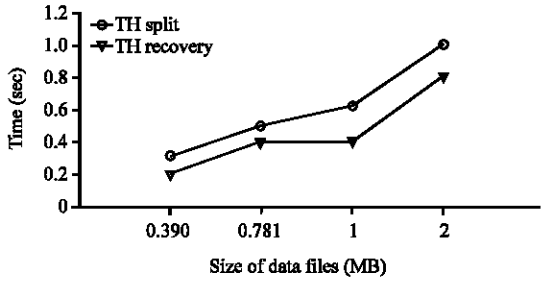

Fig. 10: Hybrid scheme split and recovery time with varied sizes |F|

security. It is comparable to Shamir's scheme, as its generated shares have high randomness, therefore, it is considered as one of the strongest secret sharing schemes where the attacker cannot get the secret by obtaining any k<m shares. Rabin's IDA has weak randomness, therefore has weak security. If an attacker gets any shares, it will be easy to recover the whole secret. The performance aspect represented in the total storage requirements for Shamir, Rabin and hybrid schemes are listed in Table 2.

As observed in Table 2 Shamir's scheme has the largest storage requirement compared with Rabin and a hybrid scheme, the size of each share is equal to size of secret. In contrast, Rabin scheme has the smallest storage requirement because the size of each share is divided by the threshold value m. Thus, Rabin's IDA is considered

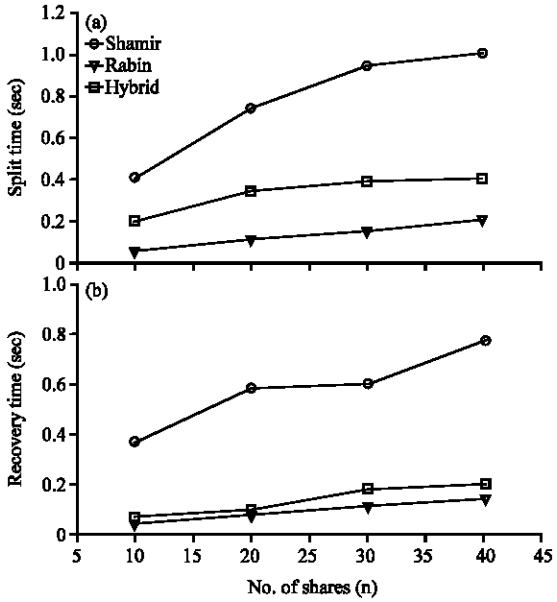

Fig. 11: a) Split and b) recover time for three schemes with varied n

Table 2: Total storage requirements for three schemes

Secret scheme	Size of shares	Total storage requirement
Shamir	|F|	n*|F|
Rabin	(|F|/m)	n*(|F|/m)
Hybrid	(|F|/m+|K|)	n*(|F|/m+|K|)

as space efficiency scheme. The storage requirement for the hybrid scheme is the same as Rabin's IDA with the added size of symmetric key.

In time aspect, as shown in Fig. 11, Shamir scheme consumes the highest time for splitting and recovering data file. Compared with hybrid and Rabin schemes, this time continues to increase with increasing number of shares n. The reason behind its high time consumption is the size of each share is equal to size of data file. The time consumed by hybrid and Rabin scheme is also increasing for big values of n and their times are nearly close to each other as shown in Fig. 11a and b. In both, the size of data file is divided by threshold values while in hybrid the size of symmetric key is inserted.

Figure 12 demonstrated the time consumed for each scheme with varied values of threshold m. The hybrid and Rabin split and recovery time continues decreasing for big values of m as shown in Fig. 12a and b. Shamir scheme split time remains steady because there is no consideration for value m in Shamir's scheme procedure as in Fig. 12a. On the other hand, Shamir scheme recovery time is increased with big values of m as shown in Fig. 12b. For three schemes, the time consumed for splitting and recovering is increasing depending on the increase in the size of the data file

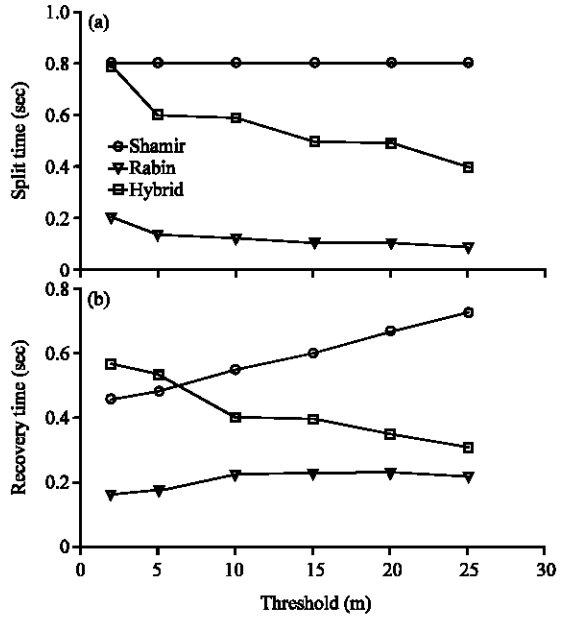

Fig. 12: a) Split and b) recover time for three schemes with varied m

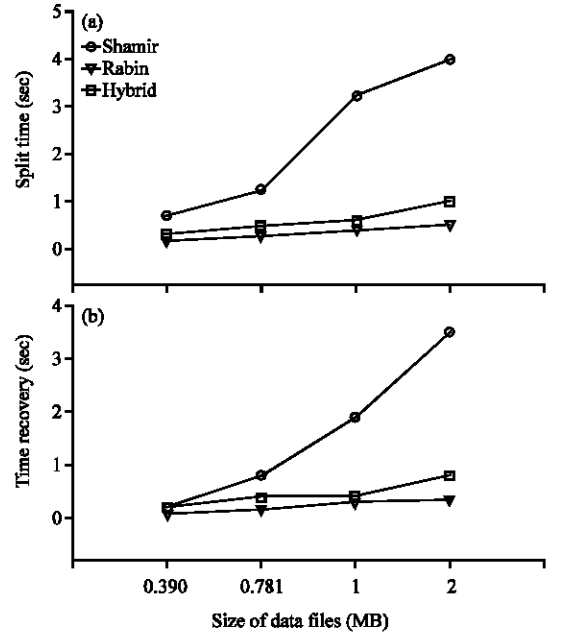

Fig. 13: a) Split and b) recover time for three schemes with varied sizes |F|

whereas Shamir's scheme is superior in this increasing the hybrid and Rabin scheme as shown in Fig. 13.

CONCLUSION

This study analyzed three secret sharing schemes, Shamir's secret sharing scheme, Rabin's IDA and hybrid secret sharing based on measured security strength, storage requirement and time consumed for splitting and recovering data. The results gained from security analysis show the robustness of Shamir's scheme and hybrid scheme while the storage requirement and time analysis shows the efficiency of Rabin's IDA.

The results of this study indicate that the hybrid scheme balanced between Rabin's IDA and Shamir's scheme in aspect of security and performance whereas it takes the security of Shamir scheme to share the key thus the storage and time costs of Shamir's scheme is avoided because the size of key is very small. Rabin's IDA is used to share data file to take advantage of its load efficiency. According to investigated results in this study we can realize and understand the current application beyond these schemes. For example, the hybrid scheme is suitable for large scale cloud computing systems because it reduces the capital investment required to store shares and ensure the security of each secret. The Shamir scheme is used to share small data size like the encryption key. This is because Shamir scheme has costs for storage requirement and time consumed. In contrast, prior to using Rabin scheme the data is encrypted because of it's a weak confidentiality, although it is more efficient in computation and storage requirements.

Future research should address Shamir's scheme, Rabin's IDA and hybrid scheme in storage distributed system such as cloud computing to analyze their security and performance in this environment.

ACKNOWLEDGEMENT

This research is supported by Ministry of Education (MOE), Malaysia and UTM under Vote No. (4L108).

REFERENCES

Annamalai, U. and K. Thanushkodi, 2013. Medical image authentication with enhanced watermarking techhnique through visual cryptography. J. Theor. Applied Inform. Technol., 57: 484-494.

Blakley, G.R., 1979. Safeguarding cryptographic keys. Proc. Natl. Comput. Conf. AFIPS., 48: 313-317.

Chor, B., E. Kushilevitz, O. Goldreich and M. Sudan, 1998. Private information retrieval. J. ACM, 45: 965-981.

Dautrich, J.L. and C.V. Ravishankar, 2012. Security Limitations of Using Secret Sharing for Data Outsourcing. In: Data and Applications Security and Privacy XXVI: 26th Annual IFIP WG 11.3 Conference, DBSec 2012, Paris, France, July 11-13, 2012. Proceedings, Cuppens-Boulahia, N., F. Cuppens and J. Garcia-Alfaro (Eds.). Springer, Heidelberg, Germany, ISBN-13: 978-3-642-31540-4, pp: 145-160.

Ermakova, T. and B. Fabian, 2013. Secret sharing for health data in multi-provider clouds. Secret sharing for health data in multi-provider clouds. Proceedings of the 15th Conference on Business Informatics, July 15-18, 2013, Vienna, pp: 93-100.

Fabian, B., T. Ermakova and P. Junghanns, 2014. Collaborative and secure sharing of healthcare data in multi-clouds. Inform. Syst., 48: 132-150.

Gentry, C., 2009. Fully homomorphic encryption using ideal lattices. Proceedings of the 41st Annual ACM Symposium on Theory of Computing, December 02-04, 2009, Seoul, Korea, pp: 169-178.

Guo, C. and C.C. Chang, 2013. A construction for secret sharing scheme with general access structure. J. Inform. Hiding Multimedia Signal Process., 4: 1-8.

Iftene, S., 2006. Secret sharing schemes with applications in security protocols. Sci. Ann. Cuza Univ., 16: 63-96.

Kantarcioglu, C. and M. Clifton, 2005. Security Issues in Querying Encrypted Data. In: Data and Applications Security XIX: 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Storrs, CT, USA, August 7-10, 2005. Proceedings, Jajodia, S. and D. Wijesekera (Eds.) Springer, Heidelberg, Germany, ISBN-13: 978-3-540-31937-5, pp: 325-337.

Krawczyk, H., 1993. Distributed fingerprints and secure information. Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Computing, August 15-18, 1993, Ithaca, New York, USA., pp: 207-218.

Krawczyk, H., 1994. Secret Sharing Made Short. In: Advances in Cryptology CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22-26, 1993, Proceedings, Stinson, D.R. (Ed.). Springer, Heidelberg, Germany, ISBN-13: 978-3-540-48329-8, pp: 136-146.

Nirmala, S.J., S.M.S. Bhanu and A.A. Patel, 2012. A comparative study of the secret sharing algorithms for secure data in the cloud. Int. J. Cloud Comput.: Serv. Archit., 2: 63-71.

Pang, L.J., H.X. Li and Y.M. Wang, 2006. A Secure and Efficient Secret Sharing Scheme with General Access Structures. In: Lecture Notes in Computer Science Fuzzy Systems and Knowledge Discovery: Third International Conference, FSKD 2006, Xi'an, China, September 24-28, 2006. Proceedings, Wang, L., L. Jiao, G. Shi, X. Li and J. Liu (Eds.). Springer, Heidelberg, Germany, ISBN-13: 978-3-540-45917-0, pp: 646-649.

Plank, J.S. and Y. Ding, 2005. Correction to the 1997 tutorial on reed-solomon coding. Software Pract. Exp., 35: 189-194.

Rabin, M.O., 1989. Efficient dispersal of information for security, load balancing and fault tolerance. J. ACM, 36: 335-348.

Resch, J.K. and J.S. Plank, 2011. AONT-RS: Blending security and performance in dispersed storage systems. Proceedings of the 9th USENIX Conference on File and Storage Technologies, February 15-17, 2011, San Jose, CA., USA., pp: 14-14.

Rogaway, P. and M. Bellare, 2007. Robust computational secret sharing and a unified account of classical secret-sharing goals. Proceedings of the 14th ACM Conference on Computer and Communications Security, October 29-November 2, 2007, Alexandria, VA, USA., pp: 172-184.

Saramentovas, A. and P. Ruzgys, 2007. Analyzing and implementing a reed-solomon decoder for forward error correction in ADSL. Aalborg University, Institute of Electronic Systems Applied Signal Processing and Implementation, Aalborg, Denmark.

Shamir, A., 1979. How to share a secret. Commun. ACM, 22: 612-613.

Slamanig, D. and C. Hanser, 2012. On cloud storage and the cloud of clouds approach. Proceedings of the International Conference for Internet Technology and Secured Transactions, December 10-12, 2012, London, pp: 649-655.

Takahashi, S., S. Kobayashi, H. Kang and K. Iwamura, 2013. Secret sharing scheme for cloud computing using Ids. Proceedings of the IEEE 2nd Global Conference on Consumer Electronics, October 1-4, 2013, Tokyo, pp: 528-529.

Venkataramana, K. and M. Padmavathamma, 2012. A threshold secure data sharing scheme for federated clouds. Int. J. Res. Comput. Sci., 2: 21-28.

Wang, C., Z. Chen, W. Yao, D. Xiao, C. Wu and J. Liu, 2010. An efficient and secure splitting algorithm for distributed storage systems. Chain Commun., 7: 89-95.