Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Application of Method *Threshold Secret Sharing* in Securing Data

Application of Method *Threshold Secret Sharing* in Securing Data

Edy Victor Haryanto*, Simeoni Daeli, Bob Subhan Riza, Juli Iriani
Faculty Of Technic And Computer Science, Universitas Potensi Utama

*edyvharyanto55@gmail.com

Abstract. *Threshold secret sharing* is one of the cryptographic techniques to secure a confidential data by dividing or distributing the data into several parts called *shares*, each part of the data does not provide any information about the secret in question if it is not combined with other parts. In general, thescheme is *secret sharing* divided into three namely *Threshold, Prevention* and *Disenrollment schemes.* The scheme applied to this application is the *threshold secret sharing. Threshold secret sharing,* has a concept that allows n people participated to hold fractional*(share)* different generated from s. Meanwhile, to reconstruct the data, it is necessary to have different pieces of*share*, each of which is held by a different participant. The security system that will be created using one of the *platforms* commonly used today is *Visual Basic .NET*
Keywords:*Secret Sharing Threshold, Data, Visual Basic .NET*

1. Introduction

Today's technological development is very rapid. With the development of technology, many people use it, especially computer technology that is needed by humans both personally and in groups (organizations) that are in dire need of computerization in storing data and every activity.

Due to the large amount of personal and confidential data from others, security is needed, especially important data in order to maintain the confidentiality of the data. In order for the data to be safe, it must be solved and stored in several places (locations). However, the problem encountered in the process of solving and storing data is a storage area that is often forgotten or even some pieces of data fall in the hands of others. And how to reunite the data or *recovery* in full. To secure the data can use Information Technology.

Threshold Secret Sharing, is a method to secure a secret message by dividing or distributing the secret message into several parts called *Share*, each part of the secret message does not provide any information about the secret in question if it is not combined with other parts. *Threshold secret sharing*, have the concept that only allows those participating to hold n fractions*(share)* different generated from s. Meanwhile, to reconstruct the data, it is necessary to have different pieces of*share*, each of which is held by a different participant. A *threshold secret sharing is* said to be ideal if *share* the resultinghas the same bit size as the size of the secret data bit. *The secret sharing threshold* has a disadvantage that is a lot of computing processes, especially for the*recovery*process. While the advantage is that if one or several of the keys are missing, then the key can still be reconstructed in full. The security system that will be created using one of the *platforms* commonly used today is *Visual Basic .NET*

2. Research Methods

2.1 Definition of Threshold Secret Sharing

Threshold is used to divide or break a secret message to 2 (two) or more recipients. (Satria Prayudi, 2015). Threshold secret sharing, can also be interpreted as to secure a secret message by dividing or distributing the message into parts called *share*.

2.2 Notation and Definition

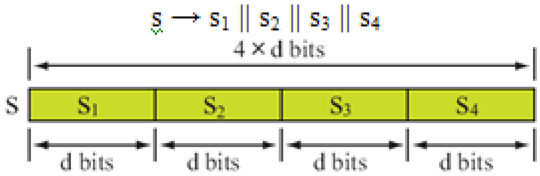
The notations used in thescheme *secret sharing* proposed by Jun Kurihara, et al. (2008), namely:

$\oplus$:operation *bitwise* XOR
|| : merging of rows of bits.
n_p : a prime number, where $n_p \geq n$.
w_i : *share* given to participant P_i, where i = 0, ..., n - 1.
s : secret message, in binary bit form, with length d * (n_p - 1) and d> 0.

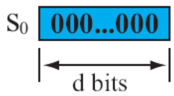
2.3 Algorithm Distribution

Message distribution algorithms using the parameters (k, n) = (3, 5) and n = n_p (Jun Kurihara, et al., 2008), can be detailed as follows:

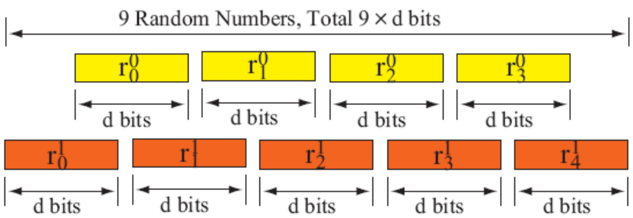
1. Solve s to be (n_p - 1) the segment with length of each d bit. In this case, n_p = 5, then n_p - 1 = 5 - 1 = 4 pieces.



2. Prepare s_0 which is a row of zero bits with d-bit length.

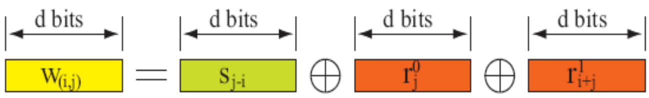


3. Generate [(k - 1) * n_p - 1] pieces of random numbers with d bits. In this case, k = 3 and n_p = 5, then [(k - 1) * n_p - 1] = [(3 - 1) * 5 - 1] = 2 * 5–1 = 9.



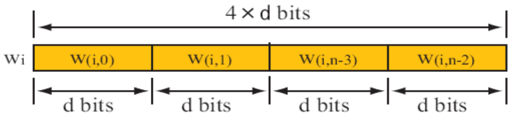
4. Execution of XOR operations with the following equation:

$$w_{(i,j)} = \left(\bigoplus_{h=0}^{k-2} r^h_{h \cdot i+j} \right) \oplus s_{j-i} = \underline{s_{j-i} \oplus r^0_j \oplus r^1_{i+j}}.$$



5. Combine n_p - 1 segment to produce the *share* w_i will be given to the participant P_i.

$$w_{(i,0)} \parallel w_{(i,1)} \parallel w_{(i,2)} \parallel w_{(i,3)} \longrightarrow w_i.$$



2.4 Algorithm Recovery

Algorithm *recovery* message using the parameters for the (k, n) = (3, 5) and n = n_p (JUN Kurihara, et al., 2008), can be seen in detail below:

1. Calculate the binary matrix G_{t0}, G_{t1}, G_{t2} so that $w_i = G_i * r$.

$$\boldsymbol{w}_i = \left(w_{(i,0)}, w_{(i,1)}, w_{(i,2)}, w_{(i,3)}\right)^{\mathrm{T}}, \quad \boldsymbol{r} = \left(r_0^0, \ldots, r_3^0, r_0^1, \ldots, r_4^1, s_1, \ldots, s_4\right)^{\mathrm{T}}$$

$$\boldsymbol{G}_i = (\boldsymbol{I}_4, \boldsymbol{E}_i, \boldsymbol{L}_{-i}).$$

2. Gauss elimination execution to obtain the M matrix.

$$\boldsymbol{G} = \left(\begin{array}{c|c} \boldsymbol{G}_{t_0} & \\ \boldsymbol{G}_{t_1} & \boldsymbol{I}_{12} \\ \boldsymbol{G}_{t_2} & \end{array}\right) = \left(\begin{array}{ccc|ccc} \boldsymbol{I}_4 & \boldsymbol{E}_{t_0} & \boldsymbol{L}_{-t_0} & 1 & & \\ \boldsymbol{I}_4 & \boldsymbol{E}_{t_1} & \boldsymbol{L}_{-t_1} & & \ddots & \\ \boldsymbol{I}_4 & \boldsymbol{E}_{t_2} & \boldsymbol{L}_{-t_2} & & & 1 \end{array}\right)$$

$$\Downarrow \textsf{Gaussian Elimination on } GF(2)$$

$$\boldsymbol{G}' = \left(\begin{array}{ccc|c} & * & & * \\ \hline \emptyset & \emptyset & \boldsymbol{I}_4 & \boldsymbol{M} \end{array}\right).$$

3. Execute the following operation to retrieve the secret:

$$(s_1, s_2, s_3, s_4) = \boldsymbol{M} \cdot \boldsymbol{w}_i$$

2.5 Message Distribution

Known secret message that you want to*share* is 'ABC' and will be formed n = 5 pieces of *shadow* and it takes k = 3 pieces of *shadow* to get the original message. The calculation process of this distribution algorithm is as follows:

1. Selected n_p = 5, then s is broken into n_p - 1 = 5 - 1 = 4 pieces.
   Message s = 'ABC' is converted to ASCII form into a Binary Code:
   0001 0100 01000010 0100 0011
   With bits = 24 bits long, broken into 4 pieces of sub-blocks, then the length of each sub-block is d = $_{24/4}$ = 6 bits. The contents of each sub-block are:
   s_1 = 010000, s_2 = 010100, s_3 = 001001, s_4 = 000011

2. Prepare s_0 = 000000
3. Generate (k - 1) * n_p - 1 = (3 - 1) * 5 - 1 = 9 pieces of random numbers r with the length of d bits.
   r (0, 0) = 010001, r (0, 1) = 001101, r (0, 2) = 100101
   r (0, 3) = 010101, r (1, 0) = 111011, r (1, 1) = 100100
   r (1, 2) = 110000, r (1, 3) = 000011, r (1, 4) = 100111

4. Perform XOR operations to calculate w (i, j):
   w (0, 0) = 101010, w (0, 1) = 111001, w (0, 2) = 000001
   w (0, 3) = 011111, w (1, 0) = 110110, w (1, 1) = 111101,
   w (1, 2) = 110110, w (1, 3) = 100110, w (2, 0) = 101000,
   w (2, 1) = 001101, w (2, 2) = 000010, w (2, 3) = 111110,
   w (3 , 0) = 000110, w (3, 1) = 100011, w (3, 2) = 011101,
   w (3, 3) = 110001, w (4, 0) = 100110, w (4, 1) = 100010,
   w (4, 2) = 001 000, w (4, 3) = 100110

Journal of Physics: Conference Series **1361** (2019) 012021 doi:10.1088/1742-6596/1361/1/012021

5. Combine n_p - 1 = 5-1 = 4 pieces ofsegment w_i to obtain the *share*wi for participants wi
   w (0) = 101010 111001 000001 011111
   w (1) = 110110 111101 110110 100110
   w (2) = 101000 001101 000010 111110
   w (3) = 000110 100011 011101 110001
   w (4) = 100110 100010 001000 100110

2.6 Recovery Message

Taken from the result of the calculation of the distribution algorithm example above, assume that the first three participants are P0, P1 and P2 want to combine their*shadow* to get the original message, then theprocess *recovery* messageis as follows:

Suppose thefirst threeselected *shadows are*:
w (0) = 101010 111001 000001 011111
w (1) = 110110 111101 110110 100110
w (2) = 101000 001101 000010 111110

So, thevalue of t_i selectedis (0, 1, 2).

Then theprocess *recovery* messageis as follows:

1. Matrix form w:

$$w = \begin{bmatrix} 101010\ 111001\ 000001\ 011111 \\ 110110\ 111101\ 110110\ 100110 \\ 101000\ 001101\ 000010\ 111110 \end{bmatrix}$$

2. Calculate the matrix vector v (t_i, j) where i = 0 to (k - 1) = (3 - 1) = 2 and j = 0 to (n_p - 2) = (5 - 2) = 3.
   The formula for determining vector values v:
   $v(t_i, j) = [i(n_p - 1, j)\ i(n_p, t_i + j)\ i(n_p, 2t_i + j) \ldots i(n_p, (k-2)t_i + j)\ i(n_p - 1, j - t_i - 1)]$
3. Form a G matrix:

$$G = \begin{pmatrix} 1000 & 10000 & 0000 \\ 0100 & 01000 & 1000 \\ 0010 & 00100 & 0100 \\ 0001 & 00010 & 0010 \\ 1000 & 01000 & 0001 \\ 0100 & 00100 & 0000 \\ 0010 & 00010 & 1000 \\ 0001 & 00001 & 0100 \\ 1000 & 00100 & 0010 \\ 0100 & 00010 & 0001 \\ 0010 & 00001 & 0000 \\ 0001 & 10000 & 1000 \end{pmatrix}$$

4. Execute *Forward Gaussian* algorithm to the matrix $[GI_{k\ (np\ -\ 1)}] = [GI_{12}]$, where I_{12} is the identity matrix with an order of 12 x 12.

$$G = \left(\begin{array}{l|l}
1000\ 10000\ 0000 & 1000\ 0000\ 0000 \\
0100\ 01000\ 1000 & 0100\ 0000\ 0000 \\
0010010000010 & 0000\ 00000000 \\
0001\ 00010\ 0010 & 0001\ 0000\ 0000 \\
1000\ 01000\ 0001 & 0000\ 1000\ 0000 \\
0100\ 00100\ 0000 & 0000\ 0100\ 0000 \\
0010\ 00010\ 1000 & 0000\ 0010\ 0000 \\
0001\ 00001\ 0100 & 0000\ 0001\ 0000 \\
1000\ 00100\ 0010 & 0000\ 0000\ 1000 \\
0100\ 00010\ 0001 & 0000\ 0000\ 0100 \\
0010\ 00001\ 0000 & 0000\ 0010\ 0010 \\
0001\ 10000\ 1000 & 0000\ 0000\ 0001
\end{array}\right)$$

Results obtained:

$$\begin{array}{c c c}
 & G_1 & \\
G_2 & \left(\begin{array}{c|c|c}
1000\ 10000 & 0000 & 1000\ 0000\ 0000 \\
0100\ 01000 & 1000 & 0100\ 0000\ 0000 \\
0010\ 00100 & 0100 & 0010\ 0000\ 0000 \\
0001\ 00010 & 0010 & 0001\ 0000\ 0000 \\
0000\ 11000 & 0001 & 1000\ 1000\ 0000 \\
0000\ 01100 & 1000 & 0100\ 0100\ 0000 \\
0000\ 00110 & 1100 & 0010\ 0010\ 0000 \\
0000\ 00011 & 0110 & 0001\ 0001\ 0000 \\
0000\ 00000 & 1011 & 0100\ 1100\ 1000 \\
0000\ 00000 & 0110 & 0110\ 1010\ 1100 \\
0000\ 00000 & 0011 & 0011\ 0101\ 0110 \\
0000\ 00000 & 0001 & 1110\ 1101\ 0011
\end{array}\right) & J_1 \\
G = & & \\
\varnothing & & J_0 \\
 & G_0 &
\end{array}$$

5. Execution algorithms *Backward Gaussian* to the matrix G_0 and J

$$\left(\begin{array}{c|c}
1011 & 0100\ 1100\ 1000 \\
0110 & 0110\ 1010\ 1100 \\
0011 & 0011\ 0101\ 0110 \\
0001 & 1110\ 1101\ 0011
\end{array}\right)$$

Results obtained:

$$\left(\begin{array}{c|c}
1000 & 0111\ 1001\ 1110 \\
0100 & 1011\ 1001\ 0010 \\
0010 & 1101\ 1000\ 0101 \\
0001 & 1110\ 1101\ 0011
\end{array}\right) \rightarrow \textbf{Results Obtained}$$

6. Based on the results *of backward substitution*as above, the value *secret s*can be calculatedas follows:

   s_1 = 010000, s_2 = 010 100, s_3 = 001 001, s_4= 000 011
   Thus, the results obtained = 010 000 010 100 001 001 000 011

   Grouped into subblok with a size of 8 bits:

01,000,001	01,000,010	01,000,011
65	66	67
A	B	C → original message

3. Results and Test

Here are the results and piloting of the threshold secretsharing:

3.1 Messages Distribution Test

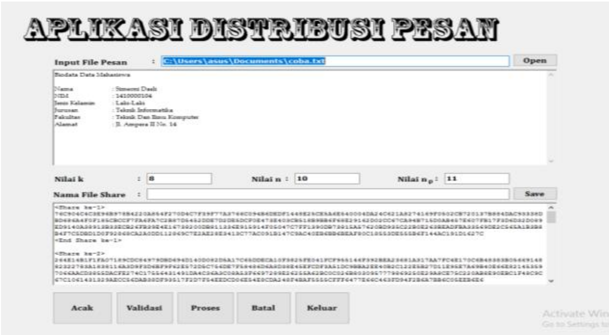


Figure 1: Threshold Secret Sharing Application

Distribution is the process of splitting or sharing of the message, so the message is safe from theft of others. The following are the results of the message distribution:

Original File
Student Biodata

Name	: Simeoni Daeli
NIM	: 1410000104
Gender	: Male
Department	: Informatics Engineering
Faculty	: Engineering and Computer Science
Address	: Jl. Ampera II No. 14

The value of k = 8 and n = 10

3.2 Results Distribution

The following are the results distribution :

<Share 1st>
76C904C4C3E96B978B4220A854F270D4C7F39F77A3766C094B6DEDF1448E25CE5A6E540004
DA24C621A8274169F0502CB720137B884DAC93338DBD686A4F0F185CBCCF7FA6FA7C2B87
D5452DDE7D2DE5DCF0E473E403CB518B9BB6F68E29162D02CC67CA94B715D0AB457E607F
B17F3D6D32D089ED9140A38913B33ECB26FB39E4E16738200DB811336E915914F05047C7FF1
390DB73815A57620BD935C22B0E263BEADFBA33569DE2C565A1B3B8B4F7C5DBD1D0F9286
8CA2A0DD112869C7E2AE28E3413C77AC091B147C9AC40EB6BB6BEAF80C18553DE555B6F1
44AC191D1627C
<End Share 1st>

Journal of Physics: Conference Series **1361** (2019) 012021 doi:10.1088/1742-6596/1361/1/012021

Recovery Book Test
The following figure shows the recovery book test of the shadow calculation results:

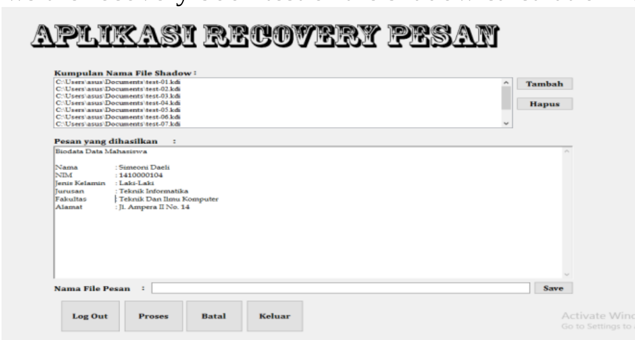


Figure 2. Shadow Calculation Results

4. Conclusions

Based on the results of the implementation and testing, the authors can conclude the following:

1. *Threshold secret sharing* is a method to secure data by dividing or distributing it into several parts
2. To do the distribution process requires the value of n and k where the value of n is the number of *shares* that you want to produce, provided that the value of n must be greater than or equal to the value of k. While the k value, which is the number of *shares* needed to get the original secret message
3. To do the*recovery*process only requires a few of the data fragments, which must correspond to the value of k or more.

References

[1]. Arya Widyadhana, Muchammad Husni, Rully Soelaiman, September 2012, **Application of *Secret Image Sharing* Using Steganography withMethods *Dynamic Embedding* and *Authentication-Chaining,*** Ten Technology Institute.

[2]. Kendall, KE and Kendall, JE, 2003, **System Analysis and Design,** Volume 1, Interpreting Thamir Abdul Hafedh Al-Hamdany, Prenhallindo, Jakarta.

[3]. Kurihara, J., Kiyomoto, S., Fukushima, K., Tanaka, T., 15 to 18 September 2008, ***A New (k, n) Threshold Secret Sharing Scheme and Its Extension,*** *11th Information Security Conference,* Taipei, KDDI R & D Laboratories, Inc., Japan.

[4]. Satria Prayudi, Robbi Rahim, 2015, **Analysis of Security on the Combination of Secret Sharing and Three-Pass Protocols,** Prima Indonesia University, Medan.

[5]. Schneier, B, 1996 ***Applied Cryptography: Protocols, Algorithms, and Source Code in C,*** *Second Edition, John Willey and Sons Inc.*

[6]. Stallings, W., 1999 ***Cryptography and Network Security: Principle and Practice,*** *second Edition, Prentice Hall.*

[7]. Zaini, March 2017, **Model of Matrix Determination Completion with Gauss Elimination Method Through Laboratory Matrix (MATLAB),** Bontang College of Technology.