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, the scheme 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 resulting has 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 the scheme secret sharing proposed by Jun Kurihara, et al. (2008), namely:

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

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

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

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

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

4. Execution of XOR operations with the following equation:
5. Combine \( n_p - 1 \) segment to produce the share \( w_i \) will be given to the participant \( P_i \).

\[
\begin{align*}
w_{i(0)} & \parallel w_{i(1)} \parallel w_{i(2)} \parallel w_{i(3)} \rightarrow w_i, \quad 4 \times d \text{ bits} \\
\end{align*}
\]

\[
\begin{array}{cccc}
w_i & W_{1(0)} & W_{1(1)} & W_{1(2,3)} & W_{1(2,0,2)} \\
d \text{ bits} & d \text{ bits} & d \text{ bits} & d \text{ bits} & d \text{ bits} \\
\end{array}
\]

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 \( G_{i0}, G_{i1}, G_{i2} \) so that \( w_i = G_i \ast r \).

\[
w_i = \begin{pmatrix} w_{i(0)}, w_{i(1)}, w_{i(2)}, w_{i(3)} \end{pmatrix}^T, \quad r = \begin{pmatrix} r_0^0, \ldots, r_3^0, r_0^1, \ldots, r_4^1, s_1, \ldots, s_4 \end{pmatrix}^T
\]

\[
G_i = \begin{pmatrix} I_4, E_{i_1}, L_{-i_1} \end{pmatrix}
\]

2. Gauss elimination execution to obtain the \( M \) matrix.

\[
G = \begin{pmatrix} G_{i0} & G_{i1} & I_{12} \\
G_{i0} & G_{i1} & 1 \\
G_{i0} & 1 \\
1 & 1 \\
\end{pmatrix} = \begin{pmatrix} I_4 & E_{i_1} & L_{-i_1} & 1 \\
I_4 & E_{i_1} & L_{-i_1} & \cdots \\
I_4 & E_{i_1} & L_{-i_1} & \cdots \\
1 & 1 & \cdots & 1 \\
\end{pmatrix}
\]

\[
G' = \begin{pmatrix} \ast & \ast \\
O & I_4 & M \\
\end{pmatrix}.
\]

3. Execute the following operation to retrieve the secret:

\[
\begin{pmatrix} s_1, s_2, s_3, s_4 \end{pmatrix} = M \cdot w_i
\]

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 \( d = \frac{24}{4} = 6 \) 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 \((k - 1) \ast n_p - 1 = (3 - 1) \ast 5 - 1 = 9 \) pieces of random numbers \( r \) with the length of \( d \) bits.

   \[
   r(0, 0) = 010001, r(0, 1) = 0011101, r(0, 2) = 100101
   \]

   \[
   r(0, 3) = 010101, r(1, 0) = 1110111, r(1, 1) = 100100
   \]

   \[
   r(1, 2) = 110000, r(1, 3) = 000011, r(1, 4) = 100111
   \]

4. Perform XOR operations to calculate \( w(i,j) \):

   \[
   w(0, 0) = 101010, w(0, 1) = 111001, w(0, 2) = 000001 \\
   w(0, 3) = 0111111, w(1, 0) = 110110, w(1, 1) = 111101, \\
   w(1, 2) = 110110, w(1, 3) = 100110, w(2, 0) = 101000, \\
   w(2, 1) = 001101, w(2, 2) = 000010, w(2, 3) = 111110, \\
   w(3, 0) = 000110, w(3, 1) = 100011, w(3, 2) = 011101, \\
   w(3, 3) = 110001, w(4, 0) = 100110, w(4, 1) = 100010, \\
   w(4, 2) = 001000, w(4, 3) = 100110
   \]
5. Combine \( n_p - 1 = 5-1 = 4 \) pieces of segment \( w_i \) to obtain the share \( w_i \) for participants \( w_i \)

\[
\begin{align*}
  w(0) &= 101010\ 111001\ 000001\ 011111 \\
  w(1) &= 111010\ 111101\ 110110\ 100110 \\
  w(2) &= 101000\ 001101\ 000010\ 111110 \\
  w(3) &= 000110\ 100011\ 011111\ 110001 \\
  w(4) &= 100110\ 100010\ 001000\ 100110
\end{align*}
\]

2.6 Recovery Message

Taken from the result of the calculation of the distribution algorithm example above, assume that the first three participants are \( P_0, P_1 \) and \( P_2 \) want to combine their shadow to get the original message, then the process recovery message is as follows:

Suppose the first three selected shadows are:

\[
\begin{align*}
  w(0) &= 101010\ 111001\ 000001\ 011111 \\
  w(1) &= 111010\ 111101\ 110110\ 100110 \\
  w(2) &= 101000\ 001101\ 000010\ 111110
\end{align*}
\]

So, the value of \( t_i \) selected is \((0, 1, 2)\).

Then the process recovery message is as follows:

1. Matrix form \( w \):

\[
w = \begin{bmatrix}
  101010 & 111001 & 000001 & 011111 \\
  111010 & 111101 & 110110 & 100110 \\
  101000 & 001101 & 000010 & 111110
\end{bmatrix}
\]

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 \( v \):

\[
v(t_i, j) = [i (n_p - 1, j) i (n_p, t_i + j) i (n_p, 2t_i + j) ... i (n_p, (k - 2) t_i + j) i (n_p - 1, j - t_i - 1)]
\]

3. Form a \( G \) matrix:

\[
G = \begin{bmatrix}
  1000 & 1000 & 0000 \\
  0100 & 0100 & 0100 \\
  0010 & 0010 & 0010 \\
  1000 & 1000 & 0001 \\
  0100 & 0100 & 0000 \\
  0010 & 0010 & 1000 \\
  0001 & 0001 & 0100 \\
  1000 & 0010 & 0010 \\
  0100 & 0010 & 0001 \\
  0010 & 0001 & 0000 \\
  0001 & 1000 & 1000
\end{bmatrix}
\]
4. Execute *Forward Gaussian* algorithm to the matrix \([G_{I_{12}}] = [G_{12}]\), where \(I_{12}\) is the identity matrix with an order of 12 x 12.

\[
G = \begin{pmatrix}
1000 & 10000 & 100000 \\
0100 & 01000 & 100000 \\
0010 & 00100 & 001000 \\
0000 & 10000 & 000000 \\
0001 & 00010 & 000100 \\
1001 & 00010 & 000100 \\
1000 & 10000 & 000000 \\
0000 & 00100 & 000100 \\
0000 & 00000 & 000000 \\
0000 & 00010 & 000010 \\
0100 & 00010 & 000100 \\
1000 & 10000 & 000000 \\
0000 & 00001 & 000001 \\
0000 & 00000 & 000000 \\
0000 & 00000 & 000000
\end{pmatrix}
\]

Results obtained:

\[
G_1 = \begin{pmatrix}
1000 & 0000 & 0000 \\
0100 & 01000 & 0000 \\
0010 & 00100 & 0100 \\
0000 & 00000 & 0010 \\
0000 & 00000 & 0001 \\
0000 & 00000 & 1000 \\
0000 & 00000 & 0000 \\
0000 & 00000 & 0000 \\
0000 & 00000 & 0000 \\
0000 & 00000 & 0000 \\
0000 & 00000 & 0000 \\
0000 & 00000 & 0000 \\
0000 & 00000 & 0000
\end{pmatrix}
\]

\[
G_2 = \begin{pmatrix}
1000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000 \\
0000 & 0000 & 0000
\end{pmatrix}
\]

\[
G_3 = \begin{pmatrix}
1011 \\
0110 \\
0011 \\
0001 \\
1110 \\
1101 \\
1011 \\
0110 \\
0011 \\
0001 \\
1110 \\
1101 \\
1011 \\
0110 \\
0011
\end{pmatrix}
\]

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

\[
\begin{pmatrix}
1011 \\
0110 \\
0011 \\
0001
\end{pmatrix}
\]

Results obtained:

\[
\begin{pmatrix}
1000 \\
0100 \\
0010 \\
0001
\end{pmatrix}
\]
6. Based on the results of backward substitution as above, the value secret can be calculated as follows:

\[ s_1 = 010000, \quad s_2 = 010100, \quad s_3 = 001001, \quad s_4 = 000011 \]

Thus, the results obtained = 010 000 010 100 001 001 000 011

Grouped into subblocks with a size of 8 bits:

\[
\begin{align*}
65 & \quad 66 & \quad 67 \\
01,000,001 & \quad 01,000,010 & \quad 01,000,011 \\
\text{A} & \quad \text{B} & \quad \text{C} \quad \text{original message}
\end{align*}
\]

3. Results and Test

Here are the results and piloting of the threshold secret sharing:

3.1 Messages Distribution Test

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:

![Threshold Secret Sharing Application](image)

Figure 1: Threshold Secret Sharing Application

| Original File | Student Biodata |
|---------------|------------------|
| Name          | Simeoni Daeli    |
| NIM           | 1410000104       |
| Gender        | Male             |
| Department    | Informatics      |
| Faculty       | Engineering and Computer Science |
| Address       | Jl. Ampera II No. 14 |

The value of \( k = 8 \) and \( n = 10 \)

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

\[ s_1 = 010000, \quad s_2 = 010100, \quad s_3 = 001001, \quad s_4 = 000011 \]

Thus, the results obtained = 010 000 010 100 001 001 000 011

Grouped into subblocks with a size of 8 bits:

\[
\begin{align*}
65 & \quad 66 & \quad 67 \\
01,000,001 & \quad 01,000,010 & \quad 01,000,011 \\
\text{A} & \quad \text{B} & \quad \text{C} \quad \text{original message}
\end{align*}
\]

3. Results and Test

Here are the results and piloting of the threshold secret sharing:

3.1 Messages Distribution Test

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      |
| Faculty       | Engineering and Computer Science |
| Address       | Jl. Ampera II No. 14 |

The value of \( k = 8 \) and \( n = 10 \)
Recovery Book Test

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

![Recovery Book Test](image.png)

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 with Methods 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.