MORE SECURE NETWORK COMMUNICATION DESIGN USING DONUT ARCHITECTURE AND BLUM-GOLDWASSER PUBLIC ENCRYPTION

Semuil Tjiharjadi
Electrical Engineering Department, Maranatha Christian University, Bandung, Indonesia
E-mail: semuil.tjiharjadi@eng.maranatha.edu

ABSTRACT

This paper describes how to design a system using Donut architecture, that can divide data to several parts and then send them after encrypt them using Blum-Goldwasser algorithm. Uniquely, even parts of them are lost; the system can reconstruct them and build data completely without losing any information.

This system can be used to send data remotely to control sensitive machine that needs accurate data to perform its actions. Invalid data or incomplete data can cause problem, especially when the data have to be sent remotely, and there are hackers, bad connection and others communication problems. It needs extra effort to make sure accurate data and this system is one of the solutions to solve these kinds of problems.

The result is a system that able to encrypt, divide, send and recover data completely if the loss factor fewer than 30%. System also can detect if there are problems of data validity.

Further research is how to use this technique to divide database and then spread them to several server in spread locations and how to protect them. The system certainly has to be able to combine and recover data completely when some servers are crash and lost their data.

***Keywords**: durable, secure, encrypt, integrity*

1. INTRODUCTION

Accurate data are needed to control machines. When remotely controlling sensitive machines, it needs a system that can make sure the integrity of perfect data are received from transmission. Beside, the confidential of the data also have to place in highest level. Hacker or cracker can use the weakness of confidential aspect to study data and break them to understand how the system works, and then it is easy for them to control the machine. Hacker or cracker can steal information, change information and create false information. This situation describes the importance of confidential aspect beside the importance of integrity aspect.

Data integrity and confidential are important aspects when sending data remotely to machine. There are two other important aspects when sending remotely, they are Validity and non-repudiation. Both of these aspects detect change of data and where is the data truly come from.

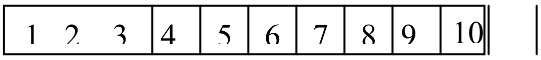
This research describes how to send remote data control that can fulfill confidential, integrity, validity and non repudiation aspects. The system have capability to reconstruct information even system loss up to 30% (parts of the loss information are not located beside it).[5]

2. DONUT ARCHITECTURE

Information file in donut architecture is divide to several parts that overlapping each others. Each part has some information that the other parts also do. This procedure makes each parts can reconstruct new information when some of the information are lost.

Figure 1. 100 KB information

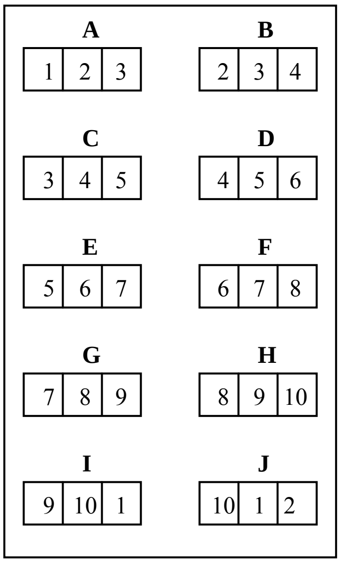
Before system sends it over network, then the data divide into several pieces, for example 100 KB information divide into 10 pieces, each piece is 10 KB as a part of 100 KB information.



1	2	3	4	5	6	7	8	9	10

Figure 2. New forms of data after it has been divided

All pieces are part of groups that will be sent partly. Each group can be constructed by 2 or more pieces. For example if each group is constructed by 3 pieces, then it will be like:



A: 1 2 3
B: 2 3 4
C: 3 4 5
D: 4 5 6
E: 5 6 7
F: 6 7 8
G: 7 8 9
H: 8 9 10
I: 9 10 1
J: 10 1 2

Figure 3. 3 Parts of information in each 10 Groups

In other way, the system can be described like a donut, each groups has some parts of data of other groups.

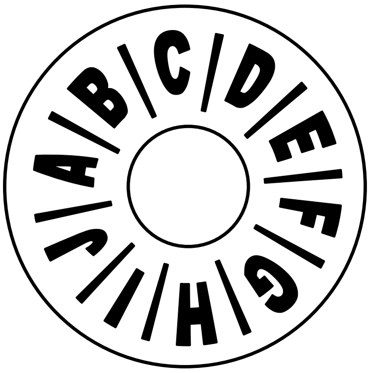


Figure 4. Donut architecture for distributed data

Donut architecture improves reconstruct ability when some of data lost and destruct integrity of their groups. Using donut architecture, system can use their part of data to develop the lost group. This can improve the integrity aspect of the whole information itself. [2]

3. CRYPTOGRAPHY

Cryptography is a study of technical mathematics that relevant with information security aspects like data validity, integrity and authentication. Or cryptography can be an art or study to keep security of a messages or information.

There are four main aspect in cryptography, they are:

1. Confidentiality, used to keep security of information from hacker or cracker.
2. Data Integrity, keeping data from illegal change of data. Keeping data integrity, system needs ability to detect manipulation like add, delete, and change of data.
3. Authentication connect with identification, sending information must be authenticated the originality and has been proved from sender and not from someone who pretend as sender.
4. Non-Repudiation, used to anticipate denial of action from sender.

3.1 Message and Encryption

A secret message is decoded by cryptography algorithm. That message is called plaintext and output of the cryptography algorithm is called ciphertext, the process converts plaintext to ciphertext is called encrypt, and the process converts ciphertext to plaintext is called decrypt.[8]

Encrypt and decrypt are mathematical transform function. If message or plaintext is symbolized as M, ciphertext is symbolized as C, encrypt process is symbolized as E, decrypt process is symbolized as D, then mathematical notation of encrypt and decrypt process will be:

Encrypt : $E(M) = C$
Decrypt : $D(C) = D(E(C)) = M$

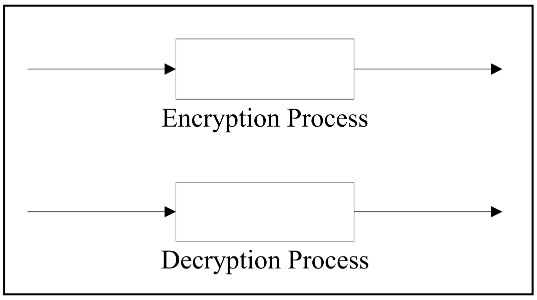


Figure 5. Encryption and Decryption process

3.2 Mathematical Theorem

Blum-Goldwasser Encryption Algorithm using some of mathematical theorems like prime numbers, modulus arithmetic operation, XOR operation exponential modulo n operation, Extended Euclidean algorithm, and quadratic residue modulo n algorithm.

3.2.1 Prime Numbers

A prime number is an integer more than one and only has 2 division factors, an integer more than one and itself. Cryptography often uses large prime numbers.

Each integer more than two can be created as:

$$n = p_1^{e1} * p_2^{e2} * p_3^{e3} * \ldots * p_k^{ek} \ ,$$

n = *integer*
p_i = *prime numbers*
e_i = *positif integer*
Example: $n = 3458 = 2 * 7 * 13 * 19$

3.2.2 Modulus Arithmetic Operation

Most computer programmers are familiar with modulus as a "remainder" operator, usually denoted by "%", which gives the remainder of an integer division instead of the quotient. For example: 27% 12 = 3

Though the idea is the same, the mechanics here are slightly different from what mathematicians refer to as modulus arithmetic. In essence, modulus arithmetic consists of taking the infinitely long number-line and coiling it around a finite circle. All the numbers that land on the same point along the circle's edge are considered interchangeable, or congruent. Thus, the analogue to the above example in modulus arithmetic would be expressed as: 27 = 3 (mod 12), or, in words: 27 is congruent to 3, modulo 12.

3.2.3 Euclidean Algorithm

Euclidean algorithm is used to find gcd (greatest common divisor). The Process of Euclidean algorithm is:

1. Chose two positive integer (a and b), and $a \geq b$

```
2. When $b \neq 0$ then do mathematical operation like:
       i.   $r \leftarrow a \bmod b$
       ii.  $a \leftarrow b$
       iii. $b \leftarrow r$
   Repeat operations until $b = 0$
3. Greatest Common Divisor (gcd) is at $a$
```

Repeating process stops until $b = 0$, and result of gcd is at a.

3.2.4 Extended Euclidean algorithm

Extended Euclidean algorithm is used to find two integers (x and y) that can fulfill this algorithm:

$ax + by = d,$
$a = integer$
$b = other\ integer$
$d = gcd(a,b)$

Extended Euclidean algorithm:

```
1. Chose two positive integer numbers ($a$ and $b$), and $a \geq b$
2. Count: $d = gcd(a,b)$
3. If $b = 0$ then:
       $d \leftarrow a$        ; $d = gcd(a,b)$
       $x \leftarrow 1$        ; $x = 1$
       $y \leftarrow 0$        ; $y = 0$
4. If $b > 0$ then process:
       i.  $q \leftarrow \lfloor a / b \rfloor$
               $r \leftarrow a - qb$
               $x \leftarrow x_2 - qx_1$
               $y \leftarrow y_2 - qy_1$
       ii. $a \leftarrow b$
               $b \leftarrow r$
               $x_2 \leftarrow x_1$
               $x_1 \leftarrow x$
               $y_2 \leftarrow y_1$
               $y_1 \leftarrow y$
   Process will repeat until $b = 0$.

5. $d \leftarrow a$            ; $d = gcd(a,b)$
   $x \leftarrow x_2$          ; $x$ = value x
   $y \leftarrow y_2$          ; $y$ = value y
```

3.2.5 Quadratic Residue Modulo n

'a' is quadratic residue modulo n when x is: $x^2 \equiv a \pmod{n}$ with condition $0 < a < n$ and n is prime number or multiply result of two prime numbers.

Quantities of quadratic residue number depend on n. If n is prime number then quantity of quadratic residue numbers are: $(n - 1) / 2$, if n is result of p multiply by q then quantity of quadratic residue numbers are $(p - 1)(q - 1) / 4$.

example: $n = 7$
$1^2 = 1 \equiv 1 \pmod{7}$
$2^2 = 4 \equiv 4 \pmod{7}$
$3^2 = 9 \equiv 2 \pmod{7}$
So *quadratic residue* modulo 7 are: { 1, 2, 4 }.

3.3 Blum-Goldwasser Encryption Method

Blum-Goldwasser encryption method was publicized first time in 1984, by its inventors, Manuel Blum and Shafi Goldwasser. Blum-Goldwasser encryption method is a public encryption, and it is using a pair of key. The first one is public key and the other is private key.

Blum-Blum-Shub generator was used to make pseudorandom bit sequence, then the process will be continued with XOR plaintext operation. The result of this process is ciphertext and it will be transmitted with pseudorandom bit by the sender.

Receiver will make pseudorandom bit sequence back using pseudorandom bit and private key, then it will process XOR operation with ciphertext to get message back.

Blum-Goldwasser encryption algorithm have 3 parts, they are: key generator algorithm, encryption algorithm, and decryption algorithm.

3.4 Key generator algorithm

Blum-Goldwasser encryption method has some keys, one public key and four private keys.

Key generator of Blum-Goldwasser encryption method process is:

- Choosing two different prime numbers randomly and each number must be congruent to *3 mod 4* operation. First number is notated as p and second number is notated as q.

  $p \equiv 3 \bmod 4$
  $q \equiv 3 \bmod 4$
- Counting p multiplying with q, and result is notated as n.

  $n = p * q$

Counting 2 integer using *Extended Euclidean algorithm*, first number is notated as a and second number is notated as b, and counting will continue until they can find the result as: $ap + bq = 1$

n is public key that will tell to sender, and p, q, a and b are private key that will be kept secretly by receiver.

3.5 Blum-Goldwasser Encryption Algorithm

Encryption process need public key (n), sequence of the process is:

- Input n
- Count $k = \log_2 n$
- Count $h = \log_2 k$
- Show sending message (*plaintext*) as sequence of *binary string*: $m = m_1 m_2 \ldots m_t$
- Chose x_0 that is a *random quadratic residue modulo n*
- Use $(i = 1 \rightarrow t)$ and count:
  - ✓ $x_i = x_{i-1} \bmod n$
  - ✓ p_i, p_i is *h least significant bits* from x_i.
  - ✓ $c_i = p_i \oplus m_i$
- Count $x_{t+1} = x_t^2 \bmod n$

- Ciphertext is sequence of c that is added by $x_{t+1;}$ $c = (c1,c2,...,ct,x_{t+1})$

3.6 Blum-Goldwasser Decryption algorithm

Decryption process needs private keys (*p*, *q*, *a* and *b*), sequence of this process is:

- Input *p*, *q*, *a* and *b*.
- Count

$$\mathrm{d}_1 = \left(\frac{(p+1)}{4}\right)^{t+1} \bmod (p-1)$$

- Count

$$\mathrm{d}_2 = \left(\frac{(q+1)}{4}\right)^{t+1} \bmod (q-1)$$

- Count u $= x_{t+1}^{d1} \bmod p$
- Count v $= x_{t+1}^{d2} \bmod q$
- Count $x_0 = (vap + ubq) \bmod n)$
- Use $(i = 1 \rightarrow t)$, and count:
  - ✓ $x_i = x_{i-1} \bmod n$
  - ✓ p_i, p_i is *h least significant bits from xi.*
  - ✓ $mi = p_i \oplus c_i$

After decryption process finishes then the output is plaintext of the message.

3.7 Program

The Program has 9 subprograms, which are:

1. Donut architecture program
2. Key generator program
3. Encryption program
4. Validation making program
5. Digital Signature program
6. Digital Signature check program
7. Decryption program
8. Validation check program
9. Data remote control system.

4. CONCLUSIONS

1. This article presented donut architecture as a good method to increase data integrity and extremely improve survival ability of data information.
2. Blum-Goldwasser encryption and decryption program able to send data that fulfill confidential factor, validity factor and integrity factor.
3. There are some unexplained characters of Blum-Characteristic Encryption and Decryption Method like a character in plaintext can be encrypted several times but will have different ciphertext. The other is size of public key in Blum-Goldwasser encryption and decryption method is not effect size of ciphertext.

REFERENCES

[1] A.Menezes, P.van Oorschot, and S.Vanstone, *Handbook of Applied Cryptography,* CRC Press, 1996.

[2] Bell, David dan Jane Frimson, *Distributed Database Systems,* Addison-Weslye Publishing Company, 1992.

[3] Douba, Salim, *Networking UNIX,* Sams Publishing,.

[4] Halvorson, Michael, *Step by Step Microsoft Visual Basic 6.0,* Microsoft Press, 1999.

[5] Ganger, Gregory R.., *Survivable Strorage System,* Carnegie Mellon University.

[6] Kusumo, Ario Suryo, *Buku Latihan Microsoft Visual Basic 6.0*, Elex Media Komputindo, 2000.

[7] Rosen, Kenneth H, Discrete Matehematics and Its Applications, 5th edition, McGraw-Hill, 2003.

[8] Schneier, Bruce, *Applied Cryptography second edition*, John Wiley & Sons, Inc., 1996.

[9] Strunk, John D., *Self-Securing Storage: Protecting Data in Compromised Systems,* Carnegie Mellon University, October 2000.Pressman, Roger S., *Software Engineering*, edisi keempat, McGraw Hill,1997.

[10] Wylie, Jay J., *Selecting the Right Data Distribution Scheme for a Survivable Strorage System*, Carnegie Mellon University, Mei 2001.