# Non Identical Record Identification

Somchai Prakancharoen
Department of computer and information science
Faculty of applied science, King mongkut's university of technology north, Bangkok, Thailand.
spk@kmutnb.ac.th

***Abstract*—This paper suggest a simple techniques that was used to generate non identical data record identification in order to increase confusion of choosing data record that kept in distributed database engines from dishonest database administrators. It was applied from cipher block chaining: CBC. This technique could be easily applied in order to enhance database security.**

***Keywords-non identical record identification;data reord:CBC***

## I. INTRODUCTION

The objective of this research is to design non identical record identification of data records that were kept in database system. The problem of data record that was kept in database engine is that there might be loosen of security from dishonest database administrators or official clients. There are some techniques that were applied in purpose of synthesis non relevance original data parts. Some of these parts should be picked up in order to compose an original plain text or original data record. After these irrelevance parts were generated they must be sent to be kept in separated database engines which were under responsible of database administrators. If these database administrators are not honest then they should conspire to consolidate all crypt records which have the same record identification in purpose of data record deliberate cryptanalysis. Hence, these crypt parts must be compiled with some technique that ease of use and hard to anticipate the parts relationship or identical.

## II. RELATED TECHNIQUES

### *A. Cipher Block Chaining: CBC [1]*

CBC is a technique that was used to increase of security in data encryption which were separated in same amount of data bits. These parts were separated encrypt with some user defined data call initialization vector (IV) which may has the same size with a being encrypt part. The result of some processing was represented as initialization vector for next iterative encryption. After whole parts encryption finished, these crypt parts were then concatenated to be a cipher text. There are many modes of CBC techniques such as Cipher block chaining, Cipher feedback, Output feedback and Counter. In case of performance of processing, Counter CBC should be suitable selection

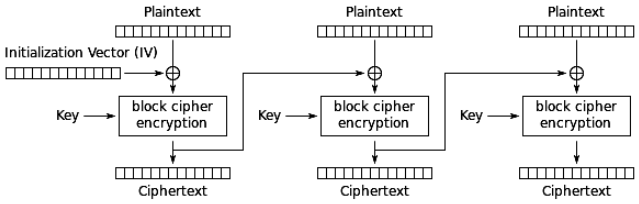


Figure 1. Cipher Block Chaining

### *B. Exclusive OR: XOR [2]*

The .XOR. is a Boolean algebra that process bit operation of two parts of binary data that have the same amount of bits. If the operands have the same value, such as both value are '1', then the .XOR. Boolean operation should return bit value of '0' but return '1' elsewhere. For example P1: '0011' and P2; '1111', the result of P1 .xor. P2 is '1100'.

### *C. Secret Sharing Scheme:SSS [3]*

SSS algorithm is mostly used in cryptography to generate the data parts which are irrelevance to a plain text. Some of these parts are then call back for mathematical computation to produce original plaintext. Many mathematical basis functions are used to reconstruct some of function values to its function formula. The most popular SSS is Shamir secret sharing algorithm which based on Lagrange basis function. Sharmir's secret sharing use n- order polynomial function. The value of functions at least n+1 value could be used to recall the original polynomial function. For example $y$ is a 2- order polynomial function. Constant $a_0$ represent plain text, $a_1$ and $a_2$ are user's arbitrary secret defined constants.

$$y = a_0 + a_1 * x + a_2 * x^2 \quad (1)$$

The value of function which are calculated from (1) then they will be assigned as irrelevance parts of data; $D_0$, $D_1$, $D_2$, $D_3$, ...$D_n$. while $D_i$ is function value on a given $x$ value. For example $D_0$ value is calculated from some $x$ value such as $x$ ='1', $a_0$ = 3, $a_1$ =2 and $a_2$ =1 then $D_0$ value is written as (1,value of function) , (1, 5) or ($x_0$, $y_0$).

$$y = 3 + 2 * 1^1 + 1 * 1^2$$
$$= 5$$

The amount of $D_i$ or 'i' must greater than n- order at least one value. These tasks were managed under a responsible of App-Ad. After App-Ad has retrieve these parts back from distributed databases then App-Ad has to reverse these parts or

function to its root polynomial function so that $a_0$ (or plain text) will be reconstructed. The reverse techniques of gathered function values back to equation (1) was based on Lagrange basis polynomial function.

$$f(x) = \sum_{j=1}^{n} y_i . l_j(x) \qquad (2)$$

While $l_j(x) = \prod_{\substack{m=0 \\ m \neq j}}^{n} \frac{x - x_m}{x_j - x_m}$

## III. Design of Non Identical Record Identification

The designed technique is presented in figure 2.

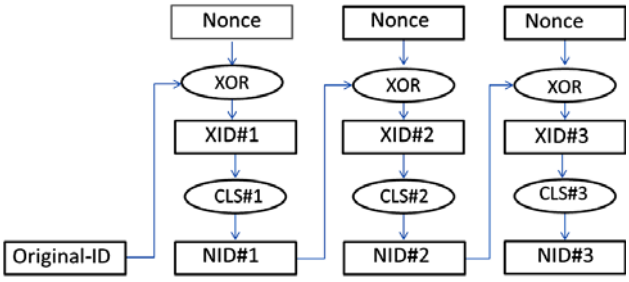


Figure 2. Non Identical Record Identification based on CBC

### A. Preparation Stage

After irrelevance data parts were generated under some secret sharing scheme , such as Sharmir's secret sharing, these parts just still has its own original record identification, called original_ID:OID. Actor – Application data owner administrator, called App-Ad, has to define 'Nonce' and amount of bits rotation left: cls#i. Nonce must have amount of bits equivalent to OID such as n bits. Nonce is arbitrary defined subject to App-Ad. CLS#i are amount of bit (in decimal) that used in circular left shift. If App-Ad want to generated three parts of data then 'i' is 3. The value of CLS#i is arbitrary defined by App_Ad.

### B. Step of Working-create

For example original-ID is '1000 0001', Nonce is '1110 0111', CLS#1=2, CLS#2=3and CLS#3=1.

$$NID\#i = << S^{cls\#i}(Nonce \oplus NID\#i-1) >>$$

$$NID\#0 = Original-ID$$

While $s^{cls\#i}$ represent rotation left on $cls\#i$ bits operation.

The 1st loop processing:

Original_ID .xor. Nonce = '1000 0001' .xor. '1110 0111'
= '0110 0110'
= XID#1

Cls#1(2) of XID#1 = '1001 1001'
= NID#1

The 2nd loop processing:

XID#1 .xor. Nonce = '1001 1001' .xor. '1110 0111'
= '0111 1110'
= XID#2

Cls#2(3) of XID#2 = '1111 0011'
= NID#2

The 3rd loop processing:

XID#2 .xor. Nonce = '1111 0011' .xor. '1110 0111'
= '0001 0100'
= XID#3

Cls#3(2) of XID#3 = '0101 0000'
= NID#3

NID#i,i=1,3 were assigned to three part of generated irrelevance data records. These parts are stamped with different record identification. These parts were sent to be kept in distributed database engines. Hence, DBA of each distributed database could not understand what the data context and what is it real record identification. DBAs could not inspect which unknown context records that were gathered from all database engines are the essential parts that should consolidate to the original data record.

### C. Recovery of Original Record Identification

In situation that App-Ad want to retrieve the specific record, App-Ad. He has to reprocess in topic "B" in order to find out all NID#i. Hence App-Ad has to kept in secret a detail of origin record identification, Nonce and, CLS#i.

### D. Secret Sharing

This paragraph is an example of designed technique that was used to generate irrelevance parts of original data record. Some of these data parts are essence to be retrieved in order to recalculate original data: plain text. Sharmir's secret sharing was used to demonstrate in this mission.

- Polynomial n- order definition

App-Ad has to arbitrarily define a polynomial equation in n-order. For example order is 2, this equation was composed of $a_0$ (plain text) = 'A' (or '65' in decimal), $a_1$= 2 and $a_2$ = 3. Assume that App-ad has defined his secret 32-order polynomial equation as equation (1).

$$y = 65 + 2x + 3x^2 \qquad (3)$$

From this 2 – order equation , App-Ad assume that k = 3 then App-Ad must generated $D_i$ at least 3 parts. In this example present that App-Ad choose 3 parts up to $x$ value '1', '2' and '3'. These $x$ values give $y$ values on '70', '81'

and '98', respectively. Hence $D_0$ = (1,70), $D_1$=(2,81) and $D_2$=(3,98).

- Lagrange basis polynomial function

$D_i$, which were stamped heading with not identical record identification, were sent to be kept in distributed database engines.

When App-Ad wants to decrypt the context of data record, he has to retrieve at least 'n +1' parts from larger than n +1 kept parts. In this example we assume 3 parts were retrieved. App-Ad has to calculate for $l_0$, $l_1$ and $l_2$ then summation them to an original polynomial equation, as (4).

$$l_0(x) = \frac{x-x_1}{x_0-x_1}.\frac{x-x_2}{x_0-x_2}$$

$$= \frac{x-2}{1-2}.\frac{x-3}{1-3}$$

$$= 3-\frac{5}{2}x+\frac{1}{2}x^2$$

$$l_1(x) = \frac{x-x_0}{x_1-x_0}.\frac{x-x_2}{x_1-x_2}$$

$$= \frac{x-1}{2-1}.\frac{x-3}{2-3}$$

$$= -3+4x-x^2$$

$$l_2(x) = \frac{x-x_0}{x_2-x_0}.\frac{x-x_1}{x_2-x_1}$$

$$= \frac{x-1}{3-1}.\frac{x-2}{3-2}$$

$$= 1-\frac{3}{2}x+\frac{1}{2}x^2$$

$$f(x) = \sum_{j=1}^{2} y_i.l_j(x)$$

$$= [70*l_0 + 81*l_1 + 98*l_2]$$

$$y = 65 + 2*x + 3*x^2 \quad (4)$$

Hence, App-Ad can easily get original plaintext ($a_0$ = 65).

## IV. Research Summary and Sugestion

This paper presents a simple technique of data record management in security database manipulation. This technique is suitable for transaction management of sensitive data record. However, the availability of file manipulation should be decreased. For performance increasing, CBC with the mode 'counter' should be examined. SSS activity is also increase computational time hence other mathematical basis function, such as Homomorphic secret sharing, that consume less computational time, should be applied in further research.

## References

[1] Murat Kantacioru, "Mode of operation CBC", Utdallas, USA, 2014.

[2] Bruce Simmons, "Binary XOR operation" , Mathword,USA, 2014.

[3] Amos beimel,"Secret sharing scheme", Ben-gurion university, Israel, 2013.