

# 1 Approach

In this article, we name the MAC scheme adopted in Cost-Effective Tag Design[] CETD-MAC.

## The Summary of This Section

- Simulated Replay Attack indicates weakness of CETD-MAC; shuffle may not change the content of input
- When shuffle does not change the content, the weakness of shifting stage: specific input has high collision rate
- A optiizmed cetd-MAC design.

## 1.1 Background of CETD-MAC and Security Definitions

**Notations of Symbols** Let  $\{0,1\}^n$  be the set of all n-bit binary strings. The set of all binary string is expressed as  $\{0,1\}^*$ . For a string  $X \in \{0,1\}^n$ ,  $|X|$  is its length in bits, and  $|X|_l = \lceil |X|/l \rceil$  is the length of X in l-bit blocks. Let  $0^l$  and  $1^l$  denote bit strings of all zeros and all ones. For a bit string X and an integer l that  $|X| \geq l$ ,  $\text{msb}_l(X)$  denotes the most significant l bits(left most l bits) of X and  $\text{lsb}_l(X)$  for least significant l bits(right most l bits) of X. For two bit string X and Y, we denote  $X\|Y$  or  $XY$  as the their concatenation. For bit string X whose length in bits is multiple of integer l, we denote X parted into l-bit sub-strings as  $X = (X[1]X[2] \dots X[n])_l$ , where  $X[1], X[2], \dots, X[n] \in \{0,1\}^l$ . The number of bits in a string of X is denoted as  $\text{len}(X)$ .

The block cipher encryption of a string X with a secret key K is denoted as  $E_K(X)$ .  $E_K(X)$  expresses the String mapping of  $\{0,1\}^n \rightarrow \{0,1\}^n$  where n is the  $\text{len}(X)$  and  $\text{len}(\text{output})$ .

**Specification of CETD-MAC** In this section, we depict the design details of CETD-MAC scheme. At the beginning we express notations required to understand CETD-MAC scheme.

**Definition of CETD-MAC Scheme** CETD-MAC scheme can be expressed as  $\text{tag} = \text{CETD-MAC}(M, \text{nonce-input})$ . The input arguments of CETD-MAC are message M and a tuple named nonce-input. The tuple nonce-input is the concatenation of the memory address of M, a counter and a random number, denoted as  $\text{nonce-input} = (\text{address}\|\text{counter}\|\text{random})$ . The length of nonce-input,  $\text{len}(\text{nonce-input})$ , is identical to the length of input to block cipher  $E_K(X)$ . The output of CETD-MAC is named tag, whose length is optional. We use Sub-BLK-No to express the value of  $\text{len}(M)/\text{len}(\text{tag})$ . One preliminary of CETD-MAC is that Sub-BLK-No should be no less than 2 to assure that swapping stage can work and 0s will be concatenated to the leftmost of input message when Sub-BLK-No is not integer. The concept of CETD-MAC is expressed in Figure ??.

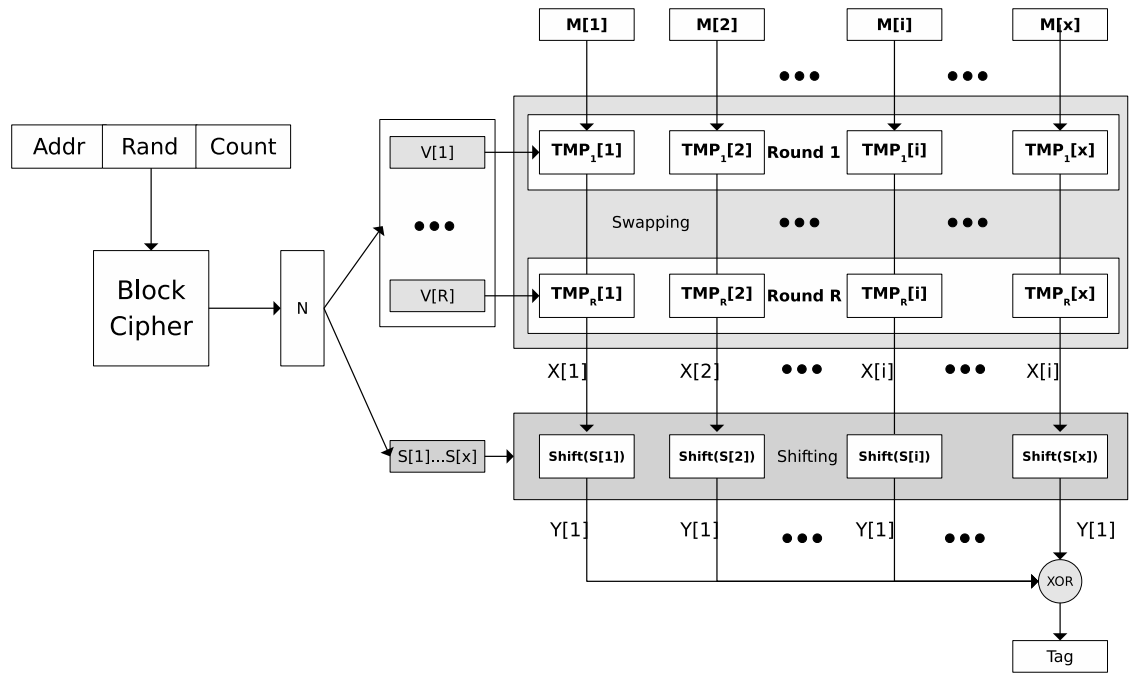


Figure 1: The CETD-MAC Scheme

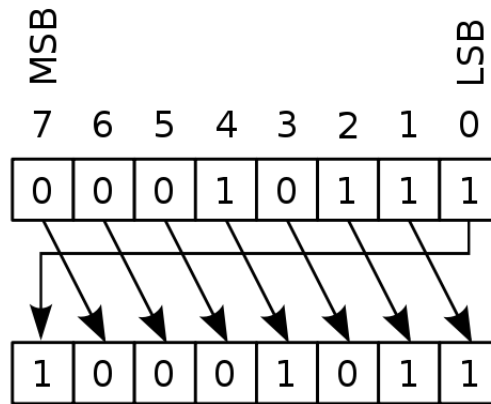


Figure 2: The Concept of Rotate Shifting(Right)

We follow the definition of CETD-MAC in [?].

- Message is splitted to sub-blocks each of whose length is tag length
- The first stage is several rounds of bit-segment swapping. For each round , two sub-blocks are randomly chosen. Introduce shuffle parameter  $V[i]$
- The output sub-blocks from swapping stage, named X blocks, are sent to block-level rotate shifting stage. Each X block is rotate shifted for several bits. The concept of rotate shifting is expressed in Fig 2 Introduce shift parameter  $S[i]$
- The output sub-blocks from shifting stage, named Y blocks, are XORed to form the final tag

As the nonce is generated by block cipher encryption  $E_K$ . Assume  $E_K$  performance like a ideal random value generator, for two distinct input nonce-input1 and nonce-input2, the corresponding nonce values N1 and N2 should be random and the probability that N1 is identical to N2 should be  $1/2^n$ , where n is  $\text{len}(N1)$ .

**Security Definitions of MAC Schemes under Replay Attacks** Our security definition of MAC schemes under replay attack follows the definition chosen-message attack. An adversary is given access to a tag generation black-box(named oracle) and a message-tag pair verification oracle. The encryption key is maintained unchanged while the nonce value is updated for each calling of tag generation oracle no matter whether the value of input message collide with a value in old timestamp. Secret key and nonces are kept in trusted area and cannot be acquired by the adversary. The adversary can copy the message-tag pairs in old time stamp. When conducting the replay attack, the adversary replaces a memory frame containing a data-tag pair in new timestamp with a copy from his storage. The forgery advantage  $F_{CETD-MAC}$  under replay attack is the probability that the adversary can get the verification oracle to accept the replace data-tag pair.

Assume the copy with old timestamp from the adversary is expressed as (M,T1) where M is the message and T1 is the corresponding tag, and the nonce of (M,T1) is denoted as N1. When applying M to the tag generation at a new time point, we mark the corresponding tag T2 and the nonce N2. If the adversary can succeed a replay attack, then T1 is identical to T2 regardless of the equality of N1 and N2. Then  $F_{CETD-MAC}$  under replay attack can be denoted as the probability  $\Pr[T1 = T2 \mid \text{Message value is M \& N1,N2 are randomly generated}]$ .

According to the design of CETD-MAC, the tag can be expressed as  $Y[1] \oplus Y[2] \oplus \dots \oplus Y[x]$ , x is the number of input sub-blocks to CETD-MAC. If the following equation is met:  $Y_1[1] \oplus Y_1[2] \dots \oplus Y_1[x] \oplus Y_2[1] \oplus Y_2[2] \dots \oplus Y_2[x] = 0$ , where  $Y_1$  blocks produce T1 and  $Y_2$  blocks generate T2, then T1 is identical to T2. This equation indicates that the behaviour of block-level rotate shifting stage effect the probability of tag collision directly. For easy understanding, we firstly analyze the

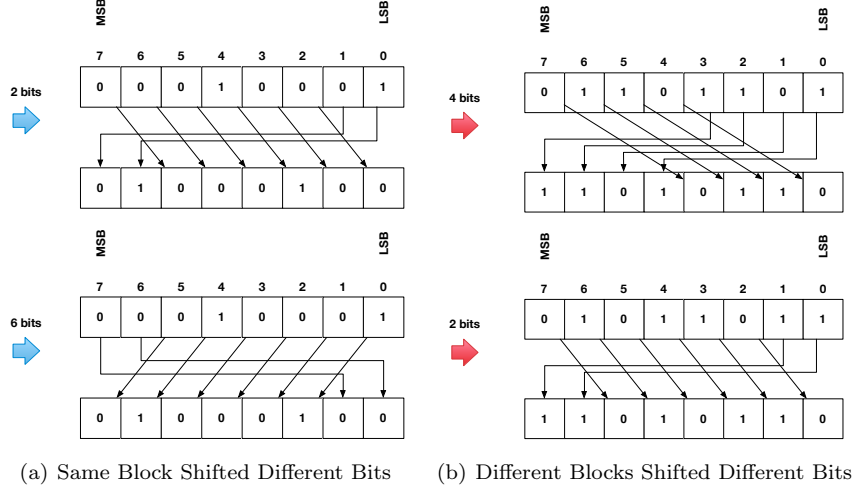


Figure 3: The Examples of Y Block Collision

case that shuffle stage does not work under replay attack, which means  $X_1[i]$  is identical to  $X_2[i]$  for any  $i \in [1, x]$ . The analysis of effect from shuffle stage on X blocks follows then.

## 1.2 Weakness of CETD-MAC

In replay attack, the adversary replace the memory frame written in new time point  $\text{Time}_2$  with a copy from old time stamp  $\text{Time}_1$ . The content in memory frame is a data-tag pair. We denote the data  $M$  and the tag in the forgery frame  $T_1$ . The output blocks from shuffle stage in generating  $T_1$  is denoted as  $X_1$  blocks and the output from shifting stage are  $Y_1$  blocks. When the forgery frame is read to be verified, the output or shuffle stage is  $X_2$  blocks and  $Y_2$  blocks for the output from shifting stage. The tag generated by verification is  $T_2$ .

In this section, we depict the weaknesses we found in CETD-MAC under replay attack. These weaknesses let the CETD-MAC to produce tags with high collision probability with some chosen inputs. The tags with highly collision probability ease the adversary's attack. Figure ?? gives the examples of the outputs from shifting stage that lead tag collision.

### 1.2.1 Weakness of Shuffle Stage

To examine the security of CETD-MAC under replay attack, we conduct a simulation experiment. The input message is consist of two blocks  $M_1$  and  $M_2$  each of whom has length of 8 bits.

We use tag distinct rate to represent the number of actual distinct tags values divided by the maximum number of possible distinct tag values. Figure ?? shows

the tag distinct rates for each input value under 3000 rounds replay attack. From Figure ?? we can find that some input values have low tag distinction rate, which means when conducting replay attack on a memory frame whose data part is in one of these inputs, the probability of tag collision is high.

Table ?? expresses the binary format of the input value resulting low tag distinction rate(below 0.1). We can see that in the two blocks of any input in the table, there is at most one bit whose value is distinct with others. According to the design of shuffle stage, in each round two blocks will do bit segment swapping to exchange bits. For input message like the instances in Table ??, the bit distribution cannot be effectly changed, which means ther is high probability that the output of shuffle stage is identical to or has the same blocks as the input message of CETD-MAC.

### 1.2.2 Weakness of Shifting Stage

**The Sequence-level Pattern Collision** Firstly, we treat blocks as a block sequences, which means in a block sequence, each block has a unique index. The definition of block sequence equality is expressed below:

**Definition 1.1.** *Sequence Equality: Two block sequences  $X1$  and  $X2$  are equal in sequence-level if  $X1[i]$  is identical to  $X2[i]$  for any  $i \in [1, x]$ .*

Assume the shuffle stage does not work, then the following fact exist:

- $X1$  and  $X2$  are two identical block sequences in sequence-level

In block-level rotate shifting stage, a segment from nonce, marked as  $S$ , is adopted as the parameter of shifting bits.  $S$  is a concatenation of sub-blocks and is denoted as  $S = (S[1] || S[2] \dots S[x])$ , where  $x$  is the number of blocks in  $X$  sequence. The value of  $S[i]$  the bits shifted for  $X[i]$  block. For two identical blocks  $X1[i]$  and  $X2[i]$ , the output blocks  $Y1[i]$  and  $Y2[i]$  are identical if  $S1[i]$  is identical to  $S2[i]$ . We found that  $Y1[i]$  and  $Y2[i]$  still have probability to be identical if  $X1[i]$  is formed by a repeated short bit segment named pattern. This property is depicted in Proposition 1.1:

**Theorem 1.1.** *Let  $X1[i]$  and  $X2[i]$  be two identical block and  $len(X1[i]) = N$ ,  $N = 2^n$ . Let  $S1[i]$  and  $S2[i]$  be two distinct shifting bit parameters for  $X1[i]$  and  $X2[i]$ . Then  $Y1[i]$  and  $Y2[i]$  can be identical only when  $X1[i]$  is formed by repeating a binary bit segment named pattern and marked as  $P$ . The length of  $P$   $len(P)$  has the following format:  $P\_L = 2^p$ ,  $p \in [0, n-1]$*

Base on Theorem 1.1, we got the following corollaries:

**Corollary 1.2.** *If a pattern  $P$  is not formed by a sub-pattern with shorter length, we call  $P$  a distinct pattern with length  $P\_L$ . The No. of distinct patterns with length  $P\_L = 2^p$  is  $2^{2^p} - 2^{2^{p-1}}$*

**Corollary 1.3.** *Assume the length of a pattern-forming  $X$  block is  $N = 2^n$ , then the No. of all possible distinct patterns is  $2^{N/2}$*

From Corollary 1.3 we can see that when the adversary conducts the replay attack on the data concatenated by pattern-formed blocks, the probability that Y block collision is high when shuffle stage does not work. The proof of Theorem 1.1, Corollary 1.2 and 1.3 is expressed in the appendix.

**The Probability of Tag Collision Under Sequence Pattern Attack** For two identical X blocks  $X\_A[i]$  and  $X\_B[i]$ , the probability that  $Y\_A[i]=Y\_B[i]$  when  $R\_A[i]$  and  $R\_B[i]$  are randomly generated is marked as  $\Pr[Y \text{ block collision}]$ .  $\Pr[Y \text{ block collision}]$  is expressed in Theorem 1.4:

**Theorem 1.4.** *Assume for two identical blocks  $X_1[i]$  and  $X_2[i]$  the length is  $N=2^n$ , and the pattern length  $P_l=2^p$  where  $p \in [0, n-1]$ . If the pattern contains no internal sub-pattern, then :*

$$\Pr[Y_1[i]=Y_2[i]] = 1/2^p \quad (1)$$

*If the pattern length in  $X[i]$  is identical for any  $i \in [1, x]$  where  $x$  is the number of blocks in X sequence, then:*

$$\Pr[Y_1[i]=Y_2[i]] = (1/2^p)^x \quad (2)$$

**The Set-level Pattern Collision** Secondly we treat blocks as a set allowing the existence of identical elements. We give the definition of block set equality below:

**Definition 1.2.** *Set Equality: Two block sets X1 and X2 are identical in set-level if the following properties are met:*

- *X1 and X2 have same number of elements*
- *The number of elements for each distinct value in X1 is identical to the number in X2*

When none of block in a X block sequence is formed by repeated pattern, it is impossible to make two Y sequences with block sequence equality by using two distinct s sequences. This assertion can be proved based on the proof of Proposition 1.1. However, we found that it is possible to form two Y sets with block set equality by using two distinct s sequences. The set-level identical Y sets can lead to tag collision directly. This attack is expressed in Proposition 1.2:

**Theorem 1.5.** *Let  $X_1[i]$  and  $X_2[j]$  be two distinct block and  $\text{len}(X_1[i])=\text{len}(X_2[j])=N$ ,  $N=2^n$ . Let  $S_1[i]$  and  $S_2[j]$  be two distinct blocks of shifting bit parameters for  $X_1[i]$  and  $X_2[j]$ . Then  $Y_1[i]$  and  $Y_2[j]$  can be identical if and only if  $X_1[i]$  can be formed by rotate shifting  $X_2[j]$ .*

Based on Theorem 1.3, we got the following corollary:

**Corollary 1.6.** *For the rotate shifting stage, when the distinct input blocks  $X_1[i]$  and  $X_2[j]$  have the property that  $X_1[i]$  can be formed by rotate shifting  $X_2[j]$ , and the shifting bits parameter  $S_1[i]$  and  $S_2[j]$  are distinct, the probability that the output blocks  $Y_1[i]$  and  $Y_2[j]$  are identical is  $Pr[Y_1[i]=Y_2[j]]=$*

Assume the shuffle stage does not work. When the adversary conducts replay attack on the memory frame whose data part is concatenated by blocks that are formed by shifting a common base block, then each element in X1 sequence can be formed by rotate shifting another element and the probability of forming identical Y sets is high, which will lead to tag collision.

**The Probability of Tag Collision Under Set Pattern Attack** Assume two identical sets X\_A and X\_B satisfy the following properties:

- Each element can be formed by rotate shifting any other element in the set
- None of the elements is formed by pattern.

As each element in the X set can be formed by rotate shifting a base block, we call such X set Same Base Block(SBB) X sets, short for SBB X sets. If none of the element is formed by a pattern, we call such X set SBB only sets.

From Corollary 1.3 we know that for a block that is N bits long, the No. of possible values that is formed by a pattern is  $2^{N/2}$ , which also means the No. of distinct values that is not formed by a pattern is  $2^{N/2}$ . Assume D\_A and D\_B are randomly generated, the probability of generating SBB only D set is expressed:

**Definition 1.3.**  $Pr[SBB\ only\ D(X)] =$

$$2^{n/2} * N^{M-1} / (2^N)^M \quad (3)$$

Using SBB only X\_A and X\_B, the probability of CIE of Y\_A and Y\_B is marked as  $Pr[CIE\ Y\ Sets]$  and expressed as the following way:

**Definition 1.4.**  $Pr[CIE\ Y\ Sets] =$

$$\prod_{i=1, j=1}^M Pr[Y\_A[i] = Y\_B[j]]$$

where M is the No. of elements in a set,  $i, j \in [1, M]$  and  $i \neq j$

For two randomly generated R\_A and R\_B, the probability that Y\_A and Y\_B are CIE when X\_A and X\_B are SBB only is expressed in the following theorem:



Figure 4: X Set Pairs with Only One Type of Y Equality

**Theorem 1.7.** Assume  $X_A$  and  $X_B$  are SBB only,  $R_A$  and  $R_B$  are randomly generated, then:

$Pr[CIE \ Y \ Sets]:$

$$\left( \sum_{K=1}^{\min(N,M)} \binom{N}{K} * (M!/v1!v2!v3! \dots vk!)^2 \right) / (N^M)^2$$

### 1.2.3 The Vulnerable Inputs of CETD-MAC Under Replay Attack

**The Case of Input Triggering Two Pattern Attacks** If the blocks in X set is formed by rotate shifting a base block and this base block is formed by a pattern, then such X set pair  $X_A$  and  $X_B$  can result either IIE or CIE Y set pairs.

Assume the  $R_A$  and  $R_B$  are randomly generated, the probability that  $Y_A$  and  $Y_B$  are SLE is expressed in the following theorem:

**Theorem 1.8.** Assume  $X_A$  and  $X_B$  are both SBB and IPL,  $R_A$  and  $R_B$  are randomly generated, then:

$Pr[SLE \ Y \ Sets]=$

$$\left( \sum_{K=1}^{\min(N,M)} \binom{N}{K} * (M!/v1!v2!v3! \dots vk!)^2 \right) / (N^M)^2$$

The proof of theorems is expressed in the appendix.

**The Behaviour of X Blocks from Shuffling Stage** In each round of Shuffle stage, two blocks from the output of previous round of shuffle are randomly



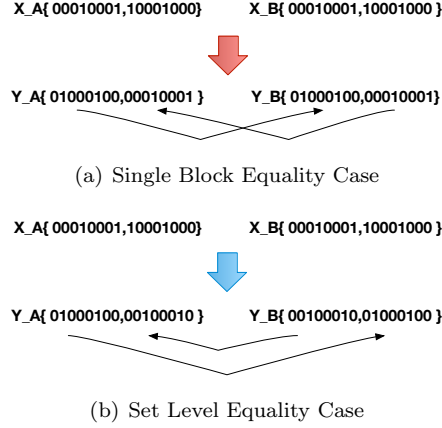


Figure 5: A X Set Pair with Two Types of Y Equality

selected then operate bit segments swapping, which will effect the distribution of bits in the two blocks. After the shuffle stage, the output, X blocks, can be classified to four categories:

1. Formed by repeated patterns
2. Formed by rotate shifting another block in this category
3. Contains the properties from categories 1 and 2
4. No properties from categories 1 or 2

Assume the block cipher used in nonce generation performances like PRF, the number of blocks in each category is effected by the input message M and the rounds of shuffle stage.

### 1.3 Security Enhanced CETD-MAC

In this section we provide an modification of original CETD-MAC. Our approach does not require additional component or information to the original CETD-MAC. We then prove that the optimized CETD-MAC can fix the weakness and protect input message from replay attack.

**Question** Using only shuffle and shift with random nonce, is it possible to ensure the range of tag to  $2^n$ ? If not, why?

## 2 Experiments and Results

### A Proof of Assertions

#### A.1 Proof of Theorem 1.1

This part proves that if two identical block  $X\_A[i]$  and  $X\_B[i]$  are shifted different bits and the result blocks remain identical,  $X\_A[i]$  is formed by pattern. The pattern length and  $\delta = |R\_A[i] - R\_B[i]|$  has such correlation:

- If  $\Delta = P\_L$  then  $Y\_A[i] = Y\_B[i]$  where  $P\_L$  the length of pattern

Assume the length of  $X\_A[i]$  is  $N$  bits, where  $N = 2^n$ . When  $X\_A[i]$  is formed by a pattern whose length is  $P\_L = 2^p$  bits, then the domain of  $Y\_A[i]$  has  $P\_L$  distinct values. There are  $N/P\_L$  distinct  $R\_A[i]$  values that shift  $X\_A[i]$  to a  $Y\_A[i]$ .

#### A.2 Proof of Theorem 1.3

If two distinct  $X$  blocks result two identical  $Y$  block. Assume the shift bits parameter blocks are  $R\_A[i]$  and  $R\_B[i]$ . Then  $X\_A[i]$  can be formed by rotate shifting  $Y\_A[i]$  for  $(N - R\_A[i]) \bmod N$  bits.  $X\_B[i]$  can be formed for  $(N - R\_B[i]) \bmod N$  bits.  $X\_B[i]$  can be formed by rotate shifting  $X\_A[i]$  for  $(R\_A[i] + N - R\_B[i]) \bmod N$  bits. Theorem 1.2 proved.

## B Computation Procedure of Probabilities

#### B.1 Proof of Theorem 1.2

For identical each block pair  $X\_A[i]$  and  $X\_B[i]$ , the No. of combination of  $R\_A[i]$  and  $R\_B[i] = N * N$ , where  $N$  is the length a  $X$  block. If  $X\_A[i]$  is formed by pattern, then  $Y\_A[i] = Y\_B[i]$  if  $|R\_A[i] - R\_B[i]| = P\_L * K$ , where  $P\_L$  is the pattern length and  $K$  is a positive integer. Then for two random  $R\_A[i]$  and  $R\_B[i]$ ,  $\Pr[Y\_A[i] = Y\_B[i] \mid X\_A[i] = X\_B[i] \text{ \& pattern} = P\_L] = \Pr[Y\_A[i] = Y\_B[i]]$  can be expressed in the following way:

- $\Pr[Y\_A[i] = Y\_B[i]] = N * (N/P\_L) / (N * N) = 1/P\_L$

#### B.2 Proof of Theorem 1.4

From Theorem 1.1 we can see that if a  $X$  block is formed by a pattern, then the No. of distinct values in the range of  $Y$  block is  $P\_L$ . While the domain of a  $R$  block has  $N$  distinct values, then for a block  $X$ , there are  $N/P\_L$  distinct  $R$  values that lead to a  $Y$  value.

If a  $X$  block is not formed by a pattern, then for each distinct  $R$  value, there is a distinct  $Y$  value. That means for a given  $X$  set that none of the blocks is

formed by a pattern, each R set will lead to a distinct Y set. The map between R set and Y set is bijection.

When each R block is randomly generated, the possible combination of R sets is  $N^M$ . That means for a given X set, the No. of possible Y set is  $N^M$ .

Assume X\_A and X\_B are IIE and each block can be formed by rotate shifting another block in the set. On the other hand, none of blocks is formed by a pattern. Then the sub-group of blocks in X\_A and X\_B can form CIE with specific R\_A and R\_B set.

Assume set Y\_A and Y\_B are SLE, then Y\_B is a permutation of Y\_A. This concept can be modeled in the following way:

- Assume the M elements in Y\_A contain K distinct values. The No. of the elements that have each value are marked as  $v_1, v_2, \dots, v_k$ .
- IF the elements in Y\_B are a permutation of the elements in Y\_A, then Y\_A and Y\_B are SLE.

Based on the basic concept in Combinatorics, the No. of a permutation of Y\_A that contains K distinct value can be expressed in the following equation:

**Definition B.1.** *No. of Permutation with K distinct Values:*

$$\binom{M}{v_1} * \binom{M-v_1}{v_2} * \binom{M-v_1-v_2}{v_3} \dots * \binom{v_k}{v_k} = M!/v_1!v_2!v_3! \dots v_k!$$

As any one of the elements in X set can be formed by rotate shifting another element, then the No. of distinct values of M elements, K, various from 1 to M. If Y\_A has K distinct values, the No. of possible combination of these K elements in Y\_A can be expressed as Com\_Y\_A:

**Definition B.2.** *Com\_Y\_A:*

$$M!/v_1!v_2!v_3! \dots v_k!$$

For each case of Y\_A in Com\_Y\_A, the No. of Y\_B to form SLE is also Com\_Y\_A. Then if Y\_A has K distinct values, Pr[CIE Y sets] can be expressed as:

**Definition B.3.** *Pr[CIE Y Sets]:*

$$\binom{N}{K} * (M!/v_1!v_2!v_3! \dots v_k!)^2 / (N^M)^2$$

If R\_A and R\_B are randomly generated, then the value of K various from 1 to M. We do th sum to get the expression in Theorem 1.4

**Definition B.4.** *Pr[CIE Y Sets]:*

$$\left( \sum_{K=1}^{\min(N,M)} \binom{N}{K} * (M!/v_1!v_2!v_3! \dots v_k!)^2 \right) / (N^M)^2$$

### B.3 Proof of Theorem 1.5

If the element contains both the properties of CIE and IIE, then the following properties:

- For two distinct sets R\_A and R\_B, Y\_A and Y\_B can be IIE
- For each value of a Y block, there are N/P\_L distinct R block that can shift X to this Y. That means for a distinct Y set, the No. of R sets is not one.
- For two distinct sets R\_A and R\_B, Y\_A and Y\_B can be CIE

Base on the Theorem 1.1, if the X block is formed by pattern, then the No. of values in the range of Y is P\_L, base on theorem 1.4, the Pr[SLE Y sets] can be expressed as:

**Definition B.5.** *Pr[SLE Y Sets]:*

$$\binom{P_L}{K} * (M! / v_1! v_2! v_3! \dots v_k!)^2 / (P_L^M)^2$$

If R\_A and R\_B are randomly generated, then the value of K varies from 1 to M. We do the sum to get the expression in Theorem 1.5

**Definition B.6.** *Pr[SLE Y Sets]:*

$$\left( \sum_{K=1}^{\min(P_L, M)} \binom{P_L}{K} * (M! / v_1! v_2! v_3! \dots v_k!)^2 / (P_L^M)^2 \right)$$