Nomenclature

$\{0,1\}^n$ Set of all n-bit binary strings

latex latex is a document

1 Approach

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

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 $\mid X \mid_l = \lceil \mid X \mid /l \rceil$ 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 $\mid X \mid \geq l$, $\text{msb}_l(X)$ 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 $X = (X[1]X[2]...X[n])_l$, where X[1], X[2], ..., $X[n] \in \{0,1\}^l$. The number of bits in a string of X is denoted as 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 len(X) and len(output).

Specification of Cost-Effective Tag Design and 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 tag = CETD-MAC(M, 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 nonce-input=(address‖counter‖random). The length of nonce-input, len(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 len(M)/len(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 **??**.

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

- Message is splitted to sub-blocks each of whose length is tag length

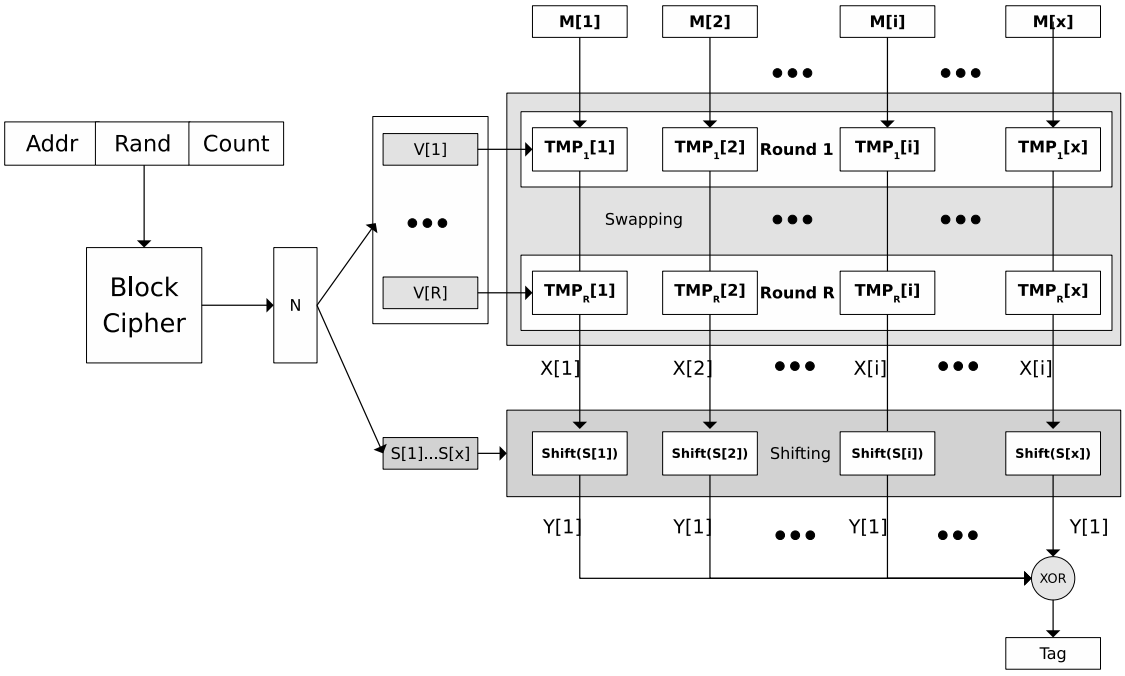


Figure 1: The CETD-MAC Scheme

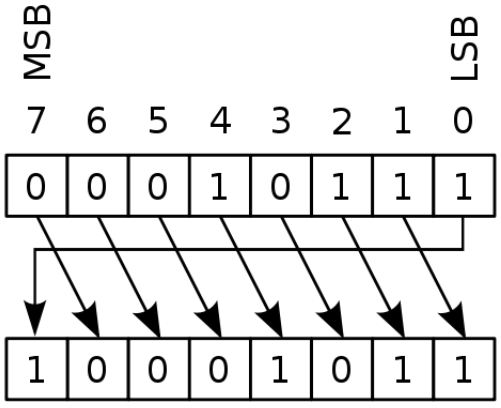


Figure 2: The Concept of Rotate Shifting(Right)

- 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 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 | Message value is M & N1,N2 are randomly generated].

According to the design of CETD-MAC, the tag can be expressed as $Y[1]\oplus Y[2]\oplus\ldots\oplus Y[x]$, x is the number of input sub-blocks to CETD-MAC. If the following equation is met: $Y_1[1]\oplus Y_1[2]\ldots\oplus Y_1[x]\oplus Y_2[1]\oplus Y_2[2]\ldots\oplus Y_2[x] = 0$, 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 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.

The Security of Cost-Effective Tag Design under Replay Attack In replay attack, the adversary replace the memory frame written in new time point Time_2 with a copy from old time stamp Time_1. The content in memory frame is a data-tag pair. To defend the replay attack, a tuple(address, counter, random number) is adopted as input to block cipher encryption to generate the nonce, which serves the control parameter for each stage in tag generation. The tuple is distinct for each data message. Assume the block cipher adopted behaves like PRF, the nonce under replay attack is random and distributed uniformly for each data message. To Cost-Effective Tag design, the security of integrity of data message replys on the security of CETD-MAC scheme.

1.2 Security Weakness of CETD-MAC under Replay Attack

As depicted in above section, if the MAC scheme is secure under replay attack, the tags should be random and distributed uniformly with repeated inputs. This property ensure the probability of the replayed data-tag pair passing verification is low. If the tags from a MAC scheme have high probability to be identical for repeated inputs, the MAC scheme is vulnerable to replay attack.

We denote the data M and the tag in the forgery frame T1. The output blocks from shuffle stage in generating T1 is denoted as X1 blocks and the output from shifting stage are Y1 blocks. When the forgery frame is read to be verified, the output or shuffle stage is X2 blocks and Y2 blocks for the output from shifting stage. The tag generated by verification is T2.

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 when replaying some chosen inputs. The tags with highly collision probability ease the adversary‘s attack.

Exposing the Weakness of CETD-MAC As the security of CETD-MAC is not provided in [], we do not know the randomness of tags under replay attack. To begin with, we conduct a simulation of replay attack on inputs formed by two 8-bit blocks concatenated. The hexadecimal value of the input various from 0x0000 to 0xFFFF. For each input, we static the tag distinction rate(TDR) under 1000 times replay attack. TDR is acquired by dividing the number of distinct values of 1000 tags with 2^8. If TDR is low, the tag collision probability is high. Figure **??** expresses the TDR of all inputs. We can see that for some specific inputs the TDR is extremely low which means the collision probability is high. The experiment results expressed in Figure **??** introduce the motivation of systematically analyzing the security of CETD-MAC under replay attack. We examined the behaviour of each stage in CETD-MAC scheme and found that each stage have a set of inputs with high probability leading output collision.

1.2.1 Weakness of Shuffle Stage

Under replay attack, the input data M is identical for the verificaiton and the replayed data-tag pair from the adversary. The basic requirement of shuffle stage is diordering the bits in M and ensure the distinction of the outputs, X blocks. If for some specific M, the X block set X1 for the verificaiton is identical to the set X2 for the adversary's data, shuffle stage is assumed lose the effect.

According to the original design of CETD-MAC, a segment of nonce is extracted as the control parameter to a round of shuffle. As nonce is the output of block cipher encryption, we assume the control parameter random. We denote this segment R[i] where is expressed the ith round of shuffle. R[i] is splitted to five bit segment components as R[i] = (index1‖offset1‖index2‖offset2‖segment length). Index determine which two blocks are selected for shuffle and offset determines the beginning bit of bit segment. The length of bit segment in block1 and block2 is determined by segment length.

We initialize the analysis of shuffle stage at 1 round shuffle. During the replay attack, when the shuffle parameters $R_1[1]$ and $R_2[1]$ is identical, the outputs of shuffle stage X1 and X2 are identical. If $R_1[1]$ and $R_2[1]$ are distinct, the equality of X1 and X2 are determined by the input M, and the two R parameters.

Assume the number of blocks in input M is x and the shuffle parameters $R_1[1]$ and $R_2[1]$ are distinct. Due to the randomness of block cipher encryption, the bit segment from nonce is assumed to be random and the value distributed uniformly. This property lead to the fact that when $R_1[1]$ and $R_2[1]$ are distinct there is at least one component pair is distinct. The analysis of the behaviour of one round shuffle is classified with each component pair. The following paragraphs in shuffle analysis is not finished, needs further discussion.

When Index Are Distinct Assume the $R_1[1]$ and $R_2[1]$ are distinct in the index components. If one pair of index is identical and the other is distinct, there are three pairs of M blocks involved in the one round shuffle, marked as ($M_1[i]$, $M_2[i]$), ($M_1[j]$,$M_2[j]$) and ($M_1[k]$,$M_2[k]$).

If both two pairs of index are distinct, there are four pairs involved in the one round shuffle, marked as ($M_1[i]$, $M_2[i]$), ($M_1[j]$,$M_2[j]$), ($M_1[k]$,$M_2[k]$) and ($M_1[h]$,$M_2[h]$).

We can see that if the blocks in a input data M involved in the shuffle has common bit segment, there is possibility to form identical X blocks. The longer the big segment is, the higher the probability that X collision occurs.

When Offset Are Distinct If the $R_1[1]$ and $R_2[1]$ are distinct in the offset components, there are two M block pairs involved, marked as ($M_1[i]$, $M_2[i]$), ($M_1[j]$,$M_2[j]$).

When Segment Length Are Distinct If the $R_1[1]$ and $R_2[1]$ are distinct in the offset components, there are two M block pairs involved, marked as ($M_1[i]$, $M_2[i]$), ($M_1[j]$,$M_2[j]$).

We can see that when the segement-length components is distinct, the X collision can be triggered with M blocks with comon bit segment.

Combination of Cases

1.2.2 Weakness of Shifting Stage

XOR Operation and Tag Collision The tag T from CETD-MAC can be expressed as the following Equation 2:

$$\mathrm{T} = \mathrm{Y}[1] \oplus \mathrm{Y}[2] \oplus \ldots \oplus \mathrm{Y}[\mathrm{x}] \tag{1}$$

Then Pr[T1 = T2] can be expressed as the Equation **??**:

$$\Pr[\mathrm{T1} = \mathrm{T2}] = \mathrm{Y}_1[1] \oplus \mathrm{Y}_1[2] \oplus \ldots \oplus \mathrm{Y}_1[\mathrm{x}] \oplus \mathrm{Y}_2[1] \oplus \mathrm{Y}_2[2] \oplus \ldots \oplus \mathrm{Y}_2[\mathrm{x}] \tag{2}$$

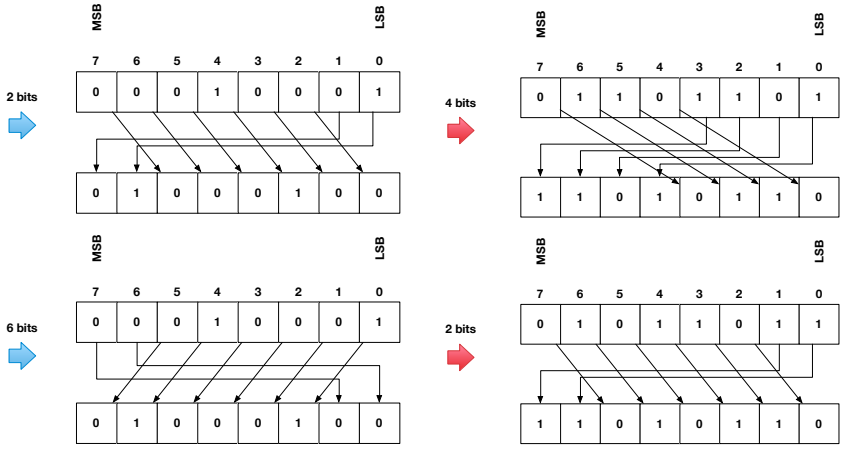
We define the concept "equivalent set" as the set whose elements are identical. We denote Y_{all} as Y1∪Y2, and Y_{sub} as a sub-set of Y_{all}. $\sum_{i=1}^{x} \oplus\ \mathrm{Y}_{sub}[\mathrm{i}]$ is denoted as $\mathrm{Y}_{sub}[1] \oplus \ldots \oplus \mathrm{Y}_{sub}[\mathrm{x}]$. Assume Y_{sub} is an equivalent set, then $\sum_{i=1}^{x} \oplus\ \mathrm{Y}_{sub}[\mathrm{i}]$ is identical to 0 if x is an even number and to $\mathrm{Y}_{sub}[1]$ if x is an odd number.

Assume Y_{all} can be splitted to several distinct equivalent Y_{sub} sets and the number of distinct Y_{sub} sets each of whom contains odd number of elements is k, then the result of $\sum_{i=1}^{2} x \oplus\ \mathrm{Y}_{all}[\mathrm{i}]$ can be expressed as $\sum_{i=1}^{k} \oplus\ \mathrm{Y}_{dist}[\mathrm{i}]$, where Y_{dist} is denoted as the set of k distinct values. The Pr[T1 = T2] is identical to $\sum_{i=1}^{k} \oplus\ \mathrm{Y}_{dist}[\mathrm{i}]$.

We can see that the probability of tag collision is determined by the number of distinct values in the union of Y1 and Y2, which are the output sets from rotate shifting stage. If the shifting stage can not ensure the randomness of Y sets if X set collision occurs , the related tags have high probability to collide. Figure **??** gives the examples of the outputs from shifting stage that lead tag collision. There are two types of repeated X sets leading to high probability of tag collision, denoted as sequence-level pattern and set-level pattern.

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 $X_1[i]$ is identical to $X_2[i]$ for any i∈[1,x].*



(a) Same Block Shifted Different Bits (b) Different Blocks Shifted Different Bits

Figure 3: The Examples of Y Block Collision

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]...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 X_1[i] and X_2[i], the output blocks Y_1[i] and Y_2[i] are identical if S_1[i] is identical to S_2[i]. We found that Y_1[i] and Y_2[i] still have probability to be identical if X_1[i] is formed by a repeated short bit segment named pattern. This property is depicted in Proposition 1.1:

Theorem 1.1. *Let X_1[i] and X_2[i] be two identical block and len(X_1[i])=N, $N=2^n$. Let S_1[i] and S_2[i] be two distinct shifting bit parameters for X_1[i] and X_2[i]. Then Y_1[i] and Y_2[i] can be identical only when X_1[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∈[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 $2^{N/2}$*

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 block collision]. Pr[Y 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 :*

$$Pr[Y_1[i]=Y_2[i]] = 1/2^p \tag{3}$$

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 Theorem 1.5:

Theorem 1.5. *Let $X_1[i]$ and $X_2[j]$ be two distinct block and $len(X_1[i])=len(X_2[j])=N$, $N=2^n$. 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 1.5, 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:*

$$Pr[Y_1[i]=Y_2[j]]= 1/n \tag{4}$$

When the adversary conducts replay attack on the memory frame whose data part is concatenated by blocks that are formated by shifting a common base block, 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. This fact is obvious if the shuffle stage cannot change the content of input data M.

The Probability of Tag Collision Under Set Pattern Attack Assume X1 and X2 are identical in sequence level and each block in X1 can be formed by rotate shifting another block in X1. From Corollary 1.6 we know that for a N-bit block, the size of domain is 2^N and half of the domain is formed by repeated pattern. This fact also indicates that half of the domain of a block is not formed with a pattern, totally $2^{N/2}$ distinct values.

Assume the X sets under replay attack, marked as X1 and X2, are identical in sequence level and none of the blocks is formed by a repeated pattern. If all the X blocks are formed by rotate shifting a base block which is not consist of a pattern, we denoted such X1 and X2 sets same base sets, short for SBS sets.

When X1 and X2 are SBS sets, the probability that Y1 and Y2 are identical in set level if shifting control paramter sets S1 and S2 are randomly generated is denoted as Pr[Y1 and Y2 are set-level equal | X1 and X2 are SBS sets] i, short for Pr[Y Sets Collision]

The lower bound of Pr[Y1 and Y2 are set-level equal | X1 and X2 are SBS sets] is expressed in Theorem 1.7.

Theorem 1.7.

$$Pr[Y\ Sets\ Collision] \geq (\sum_{K=1}^{min(N,x)} (P_n^k * 1/2 * S(x,k)^2))/(N^{2x}) \tag{5}$$

S(x,k) is the Stirling Number with x and k as inputs.

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

As summarized before, the shuffle stage is vulnerable to inputs that contains long all-0 or all-1 bit segments. After splitting a input to blocks, shuffle stage cannot effectively change the bit distribution of input blocks and the output of shuffle stage, X blocks, have big probability to be identical to the input. On the other hand, shifting stage if vulnurable to the input formed with a type of patterns. The patterns can exist in sequence level or set level. The pattern in the input of shifting stage have good chance to induce identical Y sets and directly

produce tag collisions. We can see that if the input chosen by the adversary for replay attacks is the in the intersection of vulunrable inputs of shuffle and shifting stage, none of the stages can effectively randomnize the input data M under replay attack, then there is good chance to form identical Y sets even if the nonce is generated from secure block cipher encryption.

The proof of theorems is expressed in the appendix.

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.

Inspiration As depicted above, if a input message is consisted of blocks formed with a type of pattern and the proportion of 0s and 1s in the message is extremely unbalanced, the message cannot be effectively randomized by neither the shuffle stage nor the shifting stage. That means the message has high probability to induce identical Y sets or Y sequences under replay attack. The identical Y sets or Y sequences will directly induce tag collision then succeed the replay attack.

Solution As discussed above, the reason of high probability of tag collision for some specific inputs under replay attack is that neither the shuffle or the shift stage can randomnize the input then the Y sets collide easily. The solution should be randomnize the outputs from shuffle, shift or the xor stage.

As the the outputs from shuffle and shift stage is consist of multiple blocks, all the blocks should be randomnized, otherwise the patterns existing in the input data M remains in some blocks. The cost and lattency of such a optimization of CETD-MAC is dependent on the length of input data M, which is not efficient if the input is long.

Our solution is expressed as below:

- Xor a random block to the output of XOR operation
- Assume the MSB bits of nonce are adopted as control paramters for shuffle and shifting stage, we choose LSB bits on the nonce as the random block
- The random block is denoted as Mask_Tag, and len(Mask_Tag) = len(Tag).

We denote the output of original CETD-MAC scheme in our optimized scheme T_xor and tag of optimized scheme for T, the probability of tag collision of optimized scheme is expressed as Pr[tag collision].

The computation of Pr[tag collision] is expressed below:

Pr[tag collision] = Pr[Mask_Tag1 $\oplus$ T_xor1 $\oplus$ Mask_Tag2 $\oplus$ T_xor2 =0]

=Pr[Mask_Tag1 $\oplus$ Mask_Tag2 = k | k = T_xor1 $\oplus$ T_xor2] (6)

As nonce is the output of block cipher encryption, the big segment of nonce is assumed as a output from PRF. Then Pr[tag collision] = $1/2^{len(T)}$. This result means our optimized scheme can assure the randomness of tags under replay attack and is secure.

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 δ=|R_A[i]-R_B[i] | has such correlation:

- If Δ = 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 1.4

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] | X_A[i] = X_B[i] & pattern = P_L](shoft for 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

A.3 Proof of Theorem 1.5

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]) mod N bits. X_B[i] can be formed for (N-R_B[i]) mod N bits. X_B[i] can be formed by rotate shifting X_A[i] for (R_A[i] + N-R_B[i]) mod N bits. Theorem 1.2 proved.

A.4 Proof of Theorem 1.7

As the input sets of rotate shifting stage, X1 and X2, contain the following properties:

- X1 and X2 are identical in sequence level
- Each block in X1 can be formed by rotate shifint another block in X1 for several bits

After the shifting stage, the number of distinct block value in Y1 set, denoted as k, fall in the domain [1,min[N,x]]. For each k, Y1 and Y2 set are identical in set level if the following properties are met:

- Y2 blocks have k distinct values
- Assume the k distinct values in Y1 is formed as: value1 has v1 blocks, value2 has v2 blocks, ..., valuek has vk block and $\sum_{i=1}^{k}$ vi =x. Then the distribution of k values amount x blocks in Y2 set is same as the distribution in Y1 sets.

For example, assume there is 2 distinct values among 4 blocks in Y1 set, then there are two types of distribution of blocks can be (1,3) or (2,2). For type (1,3) there is four different ways to distribute 4 blocks and for type (2,2) there is three ways. The number of ways to distribute 2 values to 4 blocks is 4+3=7. If the distribution in Y1 is (1,3), then Y2 set is identical to Y1 set if the distribution of blocks in Y2 is (1,3). When there are k distinct values among x blocks, the number of ways to distribute k values to x blocks is know as Stirling Number of Second Type, marked as S(x,k). Assume there are k_x types of distribution and the number of ways to distribute k values for each distribution type is denoted as V_1,V_2,...,V_{k_x}. Then $\sum_{i=1}^{k\_x} V_i$ = S(x,k)

For each type of Y1 blocks distribution whose distribution ways number is V_i, the number of Y2 sets that are identical to Y1 is V_i. Then for each type of distribution, the probability that Y1 and Y2 set are identical in set level can be expressed as:

$$V_i^2/(N^x)^2 \tag{7}$$

Then the probability that if there are k distinct values in Y1 set Y1 and Y2 are identical in set level can be expressed as:

$$P_N^k * \sum_{i=1}^{k\_x} V_i^2/(N^x)^2 \tag{8}$$

As S(x,k) = $\sum_{i=1}^{k\_x} V_i$, $\sum_{i=1}^{k\_x} V_i^2$ is no less than 1/2 * S(x,k)^2. Then the following inequation stands:

$$P_N^k * \sum_{i=1}^{k\_x} V_i^2/(N^x)^2 \geq P_N^k * 1/2 * S(x,k)^2/(N^x)^2 \tag{9}$$

As the domain of k is [1,min(x,N)], then:

$$Pr[Y1 = Y2 \mid S1 \text{ and } S2 \text{ are random}] = (\sum_{k=1}^{min(x,N)} (P_N^k * 1/2 * S(x,k)^2))/(N^x)^2 \tag{10}$$

Theorem 1.7 is proved.