1 The Security Analysis of Cost-Effective Tag Design

In this section we analysis the security of cost-effective tag design. We firstly introduce our security evaluation model of tag design. This model splits the security analysis into two steps: scheme security and implementation security. We analysis

1.1 Threat Model of Tag Designs

The attacks discussed in this article mainly situated on the probing and tampering the data transferred between chip and external memory. In this scenario, the adversary can conduct types of attacks on the (ciphertext, tag) pair stored on the external memory or in the transformation between chip and memory. We call these two types of integrity attack content modifying attack and copying-then-replaying attack.

When conducting a content modifying attack, then adversary modifies the content of a (ciphertext, tag) pair. At least one of ciphertext and tag are distinct to the original one sent from the chip. When the chip reads the modified pair (C1, T1), this pair is verified by VF(C1, T1, ...). If VF(C1, T1, ...)=1, then the modified pair passes and the content modifying attack succeeds.

For a copying-then-replaying attack, we make the following denotion first:

- A_{origin}: The address of the external memory frame M attacked
- A_{copy}: The address of the external memory frame M1 which is copied by the adversary as the fake frame
- TS_{origin}: The generation time stamp of the external memory frame M attacked
- TS_{copy}: The generation time stamp of the external memory frame M2. M2 and M has same address while TS_{M2} is previous of TS_{M}

During a copying-then-replaying attack on a memory frame M whoes address is A_{origin} and time stamp is TS_{origin}, the adversary A replace M with a copy. This copy can be a frame from other address A_{copy} or one at same address but copied at an old time stamp TS_{copy}. The copy is a valid pair from the chip. When the chip reads the modified frame, the fake pair(C2, T2) is verified by VF(C2, T2, ...) and the copying-then-replying attack succeeds if VF(C2, T2, ...)=1.

If the fake pair from the adversary passes the verification, the integrity of the data is broken. To defend content modification attack, the MAC scheme should ensure that when the secret information used in tag generation collides for two distinct input data, the related two tags should have low probability to be identical. For copying-then-replaying attack, identical input data should have low probability to lead to tag collision if the secret information used in tag generation are distinct.

1.2 Implementation Security of Cost-Effective Tag Design

As the MAC scheme adopted in cost-effective tag design(CETD) is a statefule scheme, the input of nonce generation should be distinct for any two write-read period. This is the security requirement in implementation level.

In CETD tag design, a input of nonce generation is a tuple(addr,ctr, rnd). The meaning of each element in the tuple is listed below:

- Addr: The address of ciphertext
- Ctr: A incrementing counter
- Rnd: A random number

1.3 Scheme Security of CETD-MAC

In this section we discuss the security of CETD-MAC under two tpyes of integrity attack. The adversary is assumed to know the length of ciphertext and tag on the memory frame he acquired.

In our security analysis of CETD-MAC scheme, the adversary A has access to the tag generation scheme and verification scheme. A can input chosen ciphertexts to the tag generation scheme and determine whether the nonce for his chosen ciphertexts are fixed or not. He can acquire the value of the tag for each his chosen ciphertext while cannot access the value of the related nonce. After analyzing several valid ciphertext-tag pairs, A sends his fake pair (C_{fake}, T_{fake}) to the verification scheme. The verification scheme generates the tag T of C_{fake} and the fake pair passes the verification if T is identical to T_{fake}. Our security analysis on CETD-MAC is modeled like this:

- For content-modification attack, the adversary A fixes the nonce and inputs distinct ciphertexts to the tag generation scheme.
- For copying-then-replaying attack, the value of ciphertexts are fixed to a value C chosen by A. A inputs C continually and as the tag generation scheme to maintain the nonce distinct for each input.

After several rounds of inputing, the adversary analyses the value of tags for his chosen inputs then sends his fake pair(C_{fake}, T_{fake}) to the verification scheme.

In this section, we prove that the probability that the fake pair passes the verification scheme under both two attacks is determined by the proportion of the 1s and 0s in the input ciphertexts and provide the equations quantifying this relationship.

Chosen Message Attack In [], the authors provided a security analysis of CETD-MAC scheme. The adversary is assumed to conduct brute force attack in which he randomly chooses a ciphtertext C and a nonce N. Then he keeps C and N fixed and tries different tags until a tag T can pass the verification with C and N. All the tags tried until passing the verification form a set named tag exploration space. The author assumed a MAC to be secure if the following conditions are met:

- The size of tag exploration space is large
- If the ciphertext and nonce is randomly generated and unaccessable to the adversary, the probability to pass the verification for each tag in the tag exploration space is identical

Based the above two conditions, the author assume CETD-MAC is secure.

Goldwasser et al. introduced the concept of unforgeability of chosen message attack in []. The adversary can select numbers of valid ciphertext-tag pairs, observe the relationship between each ciphertext and its tag, then send a fake ciphertext-tag pair to the verification stags based on the observation.The adversary A1 conducting chosen-message attack is more strategical compare with the one A2 for brute-fore attack in content modification scenario, which means A1 can try less number of ciphertext-tag pairs to pass the verification. For instance, is there is a type of relationship between ciphertext and tag for a MAC scheme, A1 will realise this relationship and conduct two types of attack with high probability of passing: just modifying ciphertext from C1 to C2 and C2 has high probability to generate same tag T1; or send new pair (C2,T2) to the verification where C2 and T2 meet the relationship. In A1‘s eyes, for given ciphetext C and a unknown nonce N, the tags in the tag domain has different probability to pass the verification while the probability is identical to A2.

In this article, we assume the adversary can choose the ciphertext in both content modification and copy-then-replay attack. The adversary aims to find groups of inputs that the two inputs from same group has probability of tag collision that is much larger than $1/2^n$, where n is Len(tag).

Design Rationale of CETD-MAC CETD-MAC scheme is consist of three stages: bit-segment shuffle, cycle shift and xor. The purpose of bit-segment shuffle is to diffuse the distribution of bits in the input. The number of shuffle rounds effects the condition of diffusion. The cycle shift stage diffuses the distribution of input in each block. Another purpose of shift stage is to diffuse the blocks that do not participate the shuffle stage, which is common if the number of blocks is large while the number of shuffle rounds is small. Finally, the output blocks of shift stage are xored to form the final tag.

1.3.1 Content Modification Attack

When conducting content modification attack, the adversary modifies the content of a memory frame from (C, T) to (C1, T1). When the verification stage VF read the modified (C1, T1) pair, VF computes C1‘s tag, marked as T_{tmp}, using the nonce N which is also used in computing T. If T_{tmp} is identical to T1, then the content modification attack succeed. We can see that in content modification attack, for two pairs (C_{origin}, T_{origin}) and (C2, T1), their nonce N and N1 are identical. To succeed the content modification attack, the adversary will choose the C1 that has high probability to get tag value T In [], Hong et al. assumed that the original CETD-MAC has a broad tag exploration space and the scheme is secure under brute force attacks. The adversary is assumed to randomly choose a ciphertext and a tag then try to pass the verification stage. In real scinerio, the adversary can acquire and read the value of any memory frames he wants,

1.3.2 Replay and Relocating Attack

2 Security Improved CETD-MAC

We have shown that the original CETD-MAC cannot effectively defend two types of integrity attacks as the number of distinct tags is determined by the proportion of 1s and 0s. In this section, we propose several approaches to improve the security of CETD-MAC. Firstly we tried to utilize the nonce and the operation existing in the original CETD-MAC design and then prove the reason that this attempt cannot succeed. Then we proposed our rationale on improvement based on the instruction from [].

2.1 Improving the security of CETD-MAC with nonce and existing operations

The Operations Adopted in Original CETD-MAC

Cost and Performance Advantage of Original CETD-MAC

Why impossible to enhance security with existing operations and nonce

2.2 Using New Operations

.1 Security Theorem Proof

.1.1 Proof of Theorem ??