1 Introduction

1.1 Motivation: CETD use new designed component, the formal security is not provided with the design

Message authentication code(MAC) is a cryptographic primitive to protect the integrity of message transmitted between sender and receiver. If the content of message blocks are modified or external message blocks are injected, a secure MAC scheme should be eligible to examine the attacks. When using a MAC scheme to protect the integrity, message blocks are adopted to a MAC scheme as input and a short message block, called tag in majority of research works, is generated as output. The tag is concatenated to the input message blocks and this data-tag pair is used as unit in transmission. Early MAC schemes are symmetric-keyed, stateless(secrete key is the only resource of randomness in tag generation) and constructed with block cipher. As stateless MAC schemes suffer from replay attacks, MAC schemes using state information of message blocks protected as additional information in tag generation have been designed.

Hong, Guo and Hu [**?**]introduced a cost-effective tag design for integrity protection, specially for embedded systems. The MAC scheme adopted is an stateful design constructed with two components: bit segments swap and block-level rotate shift. When generating a tag, a nonce generated by encrypting a tuple(message address, counter, random number) is used as additional input in processing the input data for swap and shift components.

In [**?**], a security analysis of the MAC scheme adopted in is provided by indicating that tag generated is random under brute-force attack. This evaluation procedure contains the following weakness:

- The capability of the adversary is not depicited clearly
- The security result under each type of attack in the threat model is not provided
- The data processing components used in constructing the MAC scheme in cost-effective tag design are not cryptographic primitives, the security assumption of these components are unclear.
- The evaluation procedure does not belong to any common formal evaluation framework. The security result provided can not be used in comparison with the formal results from evaluation of other MAC schemes.

The tag design from Hong et al. showed reduction of on-chip, performance and memory overhead. This cost advantage attracted us to research its security with a formal approach.

1.2 Approach: Analysis the security of CETD with computational model

In this article, we will show that from the viewpoint of computational model based security analysis, the secure block cipher used in generating nonce can

not ensure the security of Cost-Effective Tag Design as a message authentication system under several attacks. To make this statement meaningful, we follow the well recognized security notions of message authentication system in computational model.

1.3 Results: found security weakness, security optimization and results for optimized design

2 Preliminary

2.1 Formal Approaches of Security Analysis

Symbolic Model based Security Analysis

Computational Sound Security Analysis based on Symbolic Model

Computational Model based Security Analysis

Code-based Game-playing Proofs

Analysis approach for the MAC scheme in Cost-Effective Tag Design

2.2 Security of MAC Schemes in Computational Model

In some research works, the MAC schemes require only secrete key to process inputs are denoted as deterministic MAC schemes. Some MAC schemes require an additional input block, named nonce, in the generation of tag. Nonce is used to provide uniqueness for each input invoking the MAC scheme. MAC schemes with nonce as additional input are called stateful MAC schemes in some research works. We use processor-memory system to explain how a MAC scheme protects the integrity of message blocks in transmission.

When the processor writes a data block D to the memory, MAC scheme computes a tag and concatenate the tag withe data forming a data-tag pair. The data-tag pair is sent to off-chip memory for storage. When the processor reads a data from memory, the data-pair is sent to the chip and to MAC scheme. Assume the data part is D and tag part is T1. MAC scheme compute a tag T2 using D T2 is compared with T1. If T1 is identical to T2, D is assumed to be untampered. Figure 3-b expresses the functionality of MAC scheme in integrity protection. If the integrity of a processor-memory system is protected by MAC scheme, the possible attacks are:

1. Modify the content of a data-tag pair
2. Insert new data-tag pair

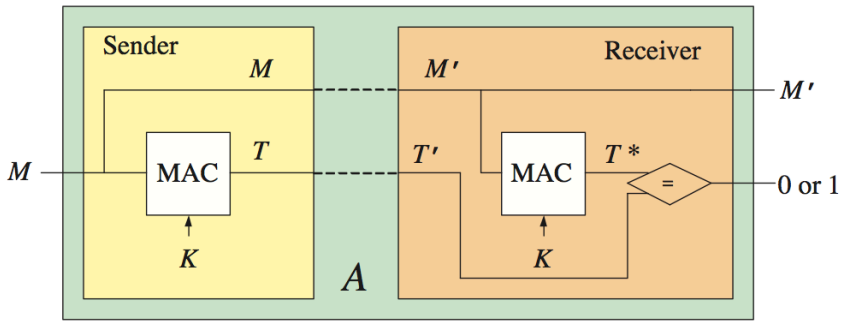

Figure 1: The Concept of Deterministic MAC Scheme

3. Replace the content of a data-tag pair with a copy of data-tag pair from other address

4. Replace the content of a data-tag pair with a copy of this pair in an old time point

If a MAC scheme is capable to defend the 1st and 2nd attacks in the list, it should ensure that for any data block D, the tag is randomly assigned. If this scheme is capable to defend the 3rd and 4th attacks too, it should ensure for any two identical data blocks D1 and D2 that are written to memory on different time or to different addresses, their tags T1 and T2 should be randomly generated.

Security Notion of MAC Schemes According to our knowledge, there have been three categories of security evaluation mechanisms designed, namely computational model based on provable security theory, dolev-yao model based on formal methods theory and automatic security analysis framework combining the advantages from previous two models.

Security Notion of Message Authentication Systems in Computational Model From the viewpoint of computational model, the security of a message authentication system can be broken If the resources, such as computation time, is unlimited for the attacker, and AP system can always be broken at some time. Research works on security analysis on AP system try quantify

the evaluation result under the assumption that the attack has limitation on computation resources.

Code-based Game-Playing Techniques In cryptography, a viewpoint of game-playing proofs represents the technique that abstracts the interaction between the adversary and environment to a program named game. Computing the probability of adversary becomes the stepwise refinement of a sequences of games. The application of this viewpoint of game-playing proofs in security analysis began with the article "Probabilistic encryption" from Goldwasser and Micali[**?**], and "Theory and applications of trapdoor functions" from Yao [**?**]and then adopted in the security analysis of various encryption systems[**?**].

In "How to protect DES against exhaustive key search", Kilian and Rogaway modeled a game with diciplines to a piece of code and introduced the idea of code-base game-playing proofs, which has been applied in encryption and authentication systems analysis.

- The benifits of modeling a game to code is xxxxxxxx.
- The procedure of applying code-based game-play in analyzing security of message authentication system
-

Cost-Effective Tag Design