# 1 Related Works

## 1.1 Preliminaries

### 1.1.1 Motivation: Data Integrity Threat Model for Processor-Memory Architecture

**What to decite in this subsection** What is data integriy for processor-memory system, how to attack the integrity

In the scenario of processor-memory architecture, data integrity refers to maintaining the data on the transmission outside the chip and stored on off-chip memory untampered. If the data stored on off-chip memory , or some external data injected to the system S, we can say that the data integrity of S is attacked. If the system S can not examine the tampered data or injected data, the attack is assumed to be successful. According to our knowledge, there are two aspects of attacks on data integrity:

- Introducing new content: The content of data in the system is modified, or data construct by the attacker is inserted to the system.
- Replacing data with valid copy: The data D in transmission or on the memory is replace with the following two types of copy: A copy of D at an old time point or a copy of other data from different memory address.

### 1.1.2 Cryptographic Authentication Primitives of Integrity Protection

**What to decite in this subsection** The methods to detect tampered data before read operation

The goal of integrity protection can be defined as the capability to examine whether the data to read by processor has been tampered. According to our knowledge, there are three common types of cryptographic primitives aiming to protect the data integrity of a system, which are cryptographic hash function, Message Authentication Code(MAC) scheme and block-level Added Redundancy Explicit Authentication (AREA).

**Cryptographic hash function** Q: What is the properties of hash function? if two D blocks have same value while the address and time point is distinct, what is their H value?

Hash is a short message block computed by hash function with data block D as input. When processor writes a data block to a memory address, the data block D is sent to Hash function and a Hash value H1 is computed and stored on the chip. The hash value H is stored on chip following the order of data block on the memory. The message stored on the memory from chip is data block D alone. When the processor wants to read a data block D from memory, D is sent to hash function and an hash value H2 is computed. If H1 is identical to H2, then the data block D is assumed to be untampered and read by processor.

Figure 3-a expresses functionality of hash function in integrity protection. If the integrity of a processor-memory system is protected by hash function, the possible attacks are:

- Modify the content of a data block
- Replace the content of a data block with a copy of data from other address
- Insert new data

Based on the procedure of intergrity protection using hash funciton, we can see that the hash function can effectively protect the integrity of data if its output hash values are randomly distributed. When meeting a effective hash function, the attacker can only perform attacks with bruteforce.

**MAC Scheme** Q:for each data block, should the nonce be distinctive?

Message Authentication Code(MAC), called tag in some research works, is a short message block. A MAC scheme accepts the data D and generate a tag with a secrete key K. Some MAC schemes require an additional input called nonce. 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 starage. 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 hash function, the possible attacks are:

- Modify the content of a data-tag pair
- Replace the content of a data-tag pair with a copy of data-tag pair from other address
- Insert new data-tag pair
- Replace the content of a data-tag pair with a copy of this pair in an old time point

**Block-level AREA**

#### 1.1.3 Evaluating the Security of Cryptographic Authentication Primitives

## 1.2 Authentication Primitives and Their Security

### 1.2.1 Hash Function

### 1.2.2 MAC Function

### 1.2.3 Block-level AREA

## 1.3 Message Authentication(MA) Systems

### 1.3.1 Hash Function Based MA Systems

### 1.3.2 MAC Function Based MA Systems

### 1.3.3 Block-level AREA Based MA Systems

### 1.3.4 Concepts of Message Authentication(MA) System

The purpose of message authentication, marked as MA, is to monitor the integrity of message blocks in the communication. In the scenario of processor-memory communication, MA can be defined as the process to verify that whether "the data read from memory by the processor (or by a specific application) at a given address is the data it last wrote at this address"([**?**]). The on-chip data is assumed to be inaccessible and secure, which is called trusted area; while the data in the transmission and stored in off-chip memory is considered to be accessible to attackers and vulnerable. Transmission path(such as bus) and off-chip memory are called malicious area. B

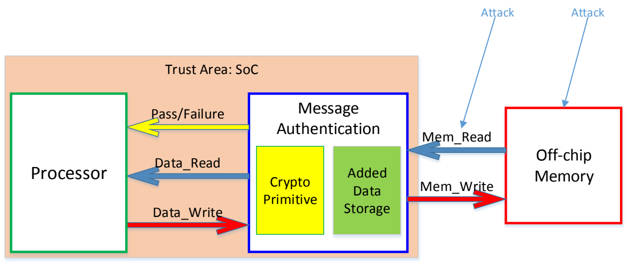

Figure 1: The Concept of Message Authentication System

Figure 1 expresses the concept of MA for processor-memory communication. When processor writes a data block, marked as D, to and address, marked as addr, on the memory, processor sends D to MA system and output a message block, marked as message(D, $\text{add}_B$), where $\text{add}_B$ represents an additional data block. The message block message(D,$\text{add}_B$) can be the data block D only, D concatenated with a short block(commonly called tag) or other format of D. The content of message(D,$\text{add}_B$) is determined by the crypographic primitive adopted by the MA system. When the processor wants to read the data D from address addr, the message(D,add$_B$) is read to SoC and sent to MA system first. After verified by MA, if the data D is not tampered, MA sends a signal indicating pass to processor and send the D to the processor, where D is extracted from message(D,add$_B$) block read from memory. If the MA finds that the D is tampered, MA sends a failure signal to processor, and invoke the exception handling procedure.

#### 1.3.5 Common Cryptographic Primitives Adopted in MA System Design

**Crypographic Hash Function** Papers proposing the theory model of hash functions

**Message Authentication Code(MAC) Function** Papers proposing MAC schemes

**Block- level Added Redundancy Explicit Authentication** Papers proposing AREA model

### 1.4 Evaluating the Security of MA System

The purpose of adopting message authentication(MA) system in the system-on-chip is protecting the integrity of the data assessed by processor. For new MA system, we expect that it is security first, then behaves good in performance and cost. So it is necessary to form security definition of MA system and do systematical evaluation of security on a new designed MA system.

**Threat Model** To our knowledge, the attacks aiming to break the integrity of data in malicious area come from two aspects:

- Introducing new data: this aspect of attack includes inserting new data block or change content of data block in malicious area
- Replacing data block with valid data: this aspect of attack includes replace a data block

#### 1.4.1 MA System Design for Memory Protection

The security of a MA system can be defined as the capability to defend the attacks in threat model.

**MA System for Uniprocessor-Memory Architecture**

**MA System for Multiprocessor-Memory Architecture**

## 1.5 Evaluating the Security of Cryptographic Primitives

As discussed above, the cryptographic primitives adopted in current posted message authentication system are cryptographic hash functions, message authentication code(MAC) functions and added redundancy explicit authentication(AREA). The security of a message authentication system

### 1.5.1 The Security of a MAC Scheme

**Message Integrity** In message transmission, the integrity of message means

### 1.5.2 Message Authentication System

A common way to protect the integrity of message blocks is utilizing message authentication system. Assume the sender A sends message M to receiver B,the message authentication system is eligible to examine the modification on M. The concept of message authentication system is expressed in Figure 1. The sender uses the message as input to the tag generation system(TG$_K$(M)) to generate a short information block called tag. The message is concatenated with tag and transmitted to the receiver. Before the receiver accept the message M, M and its tag T are sent to the verification system(VF$_K$(M,T)). If the output of verification system is 1, that means the M and T are not matched, otherwise the message M is accepted by the receiver.

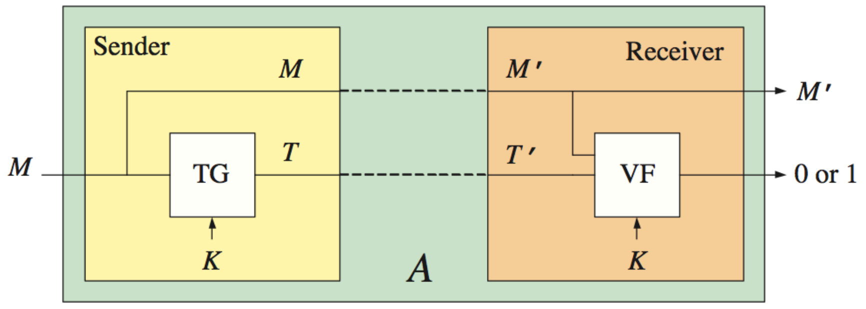

Figure 2: The Concept of MAC System

**Common Message Authentication Systems** The Message Authentication Code(MAC) is a common message authentication system. The concept of MAC scheme can be seen in Figure 2. In a deterministic MAC scheme, the verification system adopts the same key used in tag generation system. In the verification part of a MAC scheme, the tag T1 of message M is computed and compared with the tag T concatenated with the M. If T1=T then the verification system output 1 and the receiver accepts M, otherwise the verification system output 0. The early designed MAC schemes are deterministic, which means neither the sender nor the receiver needs to maintain a state used in tag generation. Latter some MAC designs adopt a state maintained by the user in the tag generation, such as the GMAC [**?**] and Cost-Effective Tag Design [**?**].

Digital signature is another kind of message authentication system. The signature generation uses a private key while the message verification stage uses public key. The digital signature system can assure non-repudiation of the message protected while MAC schemes can not.

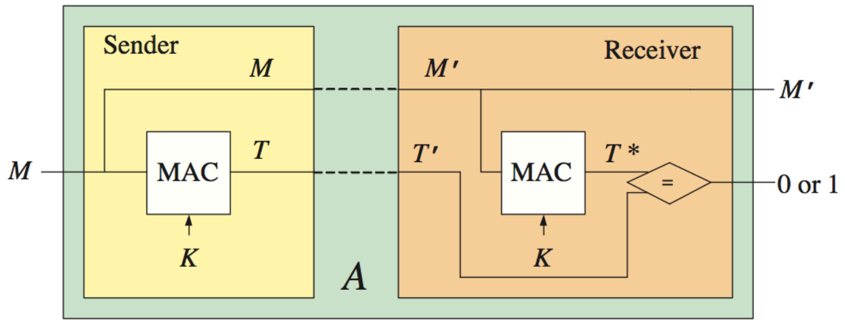

Figure 3: Message Authentication Code(MAC)

### 1.5.3 The Security of MAC schemes

**The forgery attacks** When attacking a message authentication system, the adversary try to send a pair(M,T) to the receiver to make VF$_k$(M,T)=1 while M did not originate with the legal sender. The fake pair(M$_f$,T$_f$) that makes VF$_k$(M$_f$,T$_f$)=1 is called a forgery from the adversary. A successful forgery attack indicates that the adversary has made a forgery. The purpose of a message authentication system is preventing the receiver to accept the message from unauthorized senders, such as an adversary. The quantitative property of a secure message authentication system is the low probability for an adversary to make a successful forgery attack with the limited resource.

**Chosen-message attacks** A strong type of attack that an adversary can conduct on the message authentication system is the adaptive chosen-message attack, marked as uf-cma. When doing uf-cma, the adversary chooses its own input message M and acquires the relative tag T. The adversary try to find the weakness in the design of message authentication system by analyzing the pairs(M,T) of his choice. The uf-cma provides the adversary with the most capability to succeed in the forgery attack. The probability that an adversary conducts a successful forgery attack after limited times of uf-cma is adopted as the basic quantitative security property of a message authentication in cryptography. This fact was also mentioned in [**?**].

**The Security Notions of MAC schemes** The formalised quantitative notion of the security of a MAC scheme was introduced by Bellare et al. in [**?**]. This notion follows the security notion of digital signature introduced in [**?**]. The successful forgery on a MAC scheme from an adversary A is measured by a experiment called Forgery(MAC,A). In Forgery(MAC,A), the adversary A is provided a black-box access to the tag generation system $\mathrm{TG}_K()$. When $\mathrm{TG}_K()$ takes an input message $\mathrm{M}_i$, it returns tag $\mathrm{T}_i$ to A. A conducts uf-cma by keep sending the message queries $\mathrm{M}_i$ and observes the relative tag $\mathrm{T}_i$ for limited times. On the other hand, A is provided a black-box access to the verification system $\mathrm{VF}_K()$. When A sends a pair($\mathrm{M}_j$,$\mathrm{T}_j$) to $\mathrm{VF}_K()$, the $\mathrm{VF}_K()$ computes the tag T of $\mathrm{M}_j$ and compares T with $\mathrm{T}_j$. If T=$\mathrm{T}_j$ then $\mathrm{VF}_K()$=1 otherwise 0. If A sends a pair(M,T) that makes $\mathrm{VF}_K()$ outputs 1 while M has not appeared in the previous queries of uf-cma, then A succeeds a forgery attack and Forgery(MAC,A)=1.

The quantitative security notion of a MAC scheme is forgery probability, expressed as $\mathrm{Forgery}_M AC$=Pr[Forgery(MAC,A)=1].

**The Correlation between Security and Randomness** Goldreich, Goldwasser, and Micali asserted in [**?**] that any good pseudorandom function(PRF) is a secure MAC scheme under the quantitative security notion. Bellare, Kilian and Rogaway proved this assertion in [**?**] saying that if a system behave like a pseudoranom function, this system is a secure MAC scheme if meeting the requirements on domain and range of MAC schemes. Based on these two reduction of security notion, latter researches on security evaluation of MAC schemes posted their focuses on analyzing whether the MAC scheme evaluated behaves like a PRF.

**The Randomness of a MAC scheme** The definition of PRF was introduced in [**?**] indicating that PRF could not be distinguished from a ideal random function each bit of whose output was a coin flip. To define how closely a MAC scheme behaves like a PRF, Bellare et al. provided a quantitative notion in [**?**] named Adv$_{MAC}^{PRF}$(), which was based on the concept of distinguisher introduced in[**?**].

Let F$_0$ and F$_1$ be two function with a common domain D and a common range R. A distinguisher A for F$_0$ versus F$_1$ is an adverary A that has access to a black box named oracle f:D-¿R. After accessing the oracle f, A computes a bit. Assume the function stored in the oracle f is X and A guesses that X is in the oracle, then A computes 1 otherwise 0. The the advantage of A in distinguishing F$_0$ from F$_1$ is expressed as Adv$_{F1}^{F0}$=Pr[f$\xleftarrow{R}$F0:A$^{F0}$=1]-Pr[f$\xleftarrow{R}$F1:A$^{F1}$=1]. Pr[f$\xleftarrow{R}$F0:A$^{F0}$=1] means when the content of oracle f is F0, A guesses that F0 is in oracle then output 1.

We can see that if F0 behaves much like F1, it is hard for A to distinguish between F0 and F1 then Adv$_{F1}^{F0}$ is very small. This case is adopted by Bellare et al. in the quantitative notion of randomness of a MAC scheme. If the randomness of a MAC scheme is good, then the MAC scheme behaves like a PRF and Adv$_{MAC}^{[PRF]}$ is small.

## 1.6 Implementation of Tag Design

### 1.6.1 Network and Cloud System

### 1.6.2 Memory Protection

**Single-processor System**

**Multiple-processor System**

### 1.6.3 Crypto-hardware Design

## 1.7 Security Evaluation of Tag Design