1 Related Works

1.1 Preliminaries

1.1.1 Data Integrity Threat Model for Processor-Memory Architecture

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:

1. Introducing new content: The content of data in the system is modified, or data construct by the attacker is inserted to the system.

2. 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 Protection of Integrity: Theoretical Models

The goal of integrity protection can be defined as the capability to examine whether the data to read by processor has been tampered. Cryptographic hash function and Message Authentication Code(MAC) scheme are common cryptographic primitives aiming to protect the data integrity of a system. These theoretical models for integrity protection is also called Authentication Primitives(AP) in some research works.

Cryptographic Hash Functions 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:

1. Modify the content of a data block

2. Insert new data

3. Replace the content of a data block with a copy of data from other address

If using hash function to protect integrity, the three attacks in the above list can be defended if the hash function behave like an theoretical random function, which means for any input data block D, its hash value H is randomly assigned.

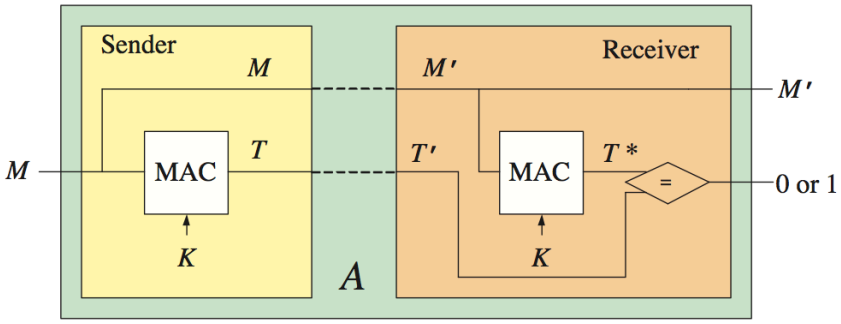

Figure 1: The Concept of Deterministic MAC Scheme

Message Authentication Code(MAC) Schemes Message Authentication Code(MAC), called tag in some research works, is a short message block. A tag is generated by a MAC scheme by accepting a data block D as input and process D with a secret key K inside the scheme. Some MAC schemes require an additional input block, named nonce, in the generation of tag.

When the processor writes a data block D to the memory, MAC scheme computes a tag and concatenate the tag withe the 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

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 writen to memory on different time or to different addresses, their tags T1 and T2 should be randomly generated.

1.2 Evaluating the Security of AP Designs

What to Say

- What is the security notion of provable, how to quatify the security of AP, the weakness
- What is the properties of formal method, how to quantify the security of AP, the weakness
- , why provable security is err-prone, why formal method is not concrete or sound
- early automative proof, as they have the weakness of formal methods, their security assumption are unrealistic
-

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. 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.

1.2.1 Computational Model based on Provable Security

Existential forgery under chosen-message attack.[**?**]. In the security notions of computational provable security, computatoin time or No. of attack rounds determine the probabiltiy of success: the adversary can always succeed if the attack rounds allowed is unlimited; otherwise the probability of success is expressed in quantity of No. of attack rounds.

In the analysis model of computational provable security, the security is quantified by the probability of successful forgery.

The forgery attacks When attacking a authentication primitive, 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 [27].

The Security Notions of MAC schemes The formalised quantitative notion of the security of a MAC scheme was introduced by Bellare et al. in [3]. This notion follows the security notion of digital signature introduced in [10]. 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 $TG_K()$. When $TG_K()$ takes an input message M_i, it returns tag T_i to A. A conducts uf-cma by keep sending the message queries M_i and observes the relative tag T_i for limited times. On the other hand, A is provided a black-box access to the verification system $VF_K()$. When A sends a pair(M_j,T_j) to $VF_K()$, the $VF_K()$ computes the tag T of M_j and compares T with T_j. If T=T_j then $VF_K()$=1 otherwise 0. If A sends a pair(M,T) that makes $VF_K()$ 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 Forgery_{MAC}=Pr[Forgery(MAC,A)=1].

The Correlation between Security and Randomness Goldreich, Goldwasser, and Micali asserted in [9] that any good pseudorandom function(PRF) is a secure MAC scheme under the quantitative security notion. Bellare, Kilian and Rogaway proved this assertion in [3] 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 [9] 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 [3] named $\mathrm{Adv}_{MAC}^{PRF}()$, which was based on the concept of distinguisher introduced in[9].

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 $\mathrm{Adv}_{MAC}^{[PRF]}$ is small.

1.2.2 Dolev-Yao Model based on Formal Methods

Dolev-Yao [**?**] framework is an approach aiming to evaluation the security of an cryptographic primitive automatically. This model is based on the princples of formal methods.

What is the properties of dolev-yao model?

1.2.3 Automated Sound Security Evaluation

Computational provable security model is sound, but the procedure is manually and informal, easy to results wrong. Formal methods can do the evaluation automatically, but the security notion is too touch, not realistic and the final result is not sound.

Research works have ben proposed that aim to design a new mechanism combining the advantages of both two models and containing no serious flaws. Early works try to introduce the security notion and concepts from computational provable security to formal method model, which will result automative security analysis whose results are sound. Add the drawbacks of these early research works.

Recent research works try to make designs from the opposite way by modifying the analysis procedure of computational provable security model to make the analysis automative. Hence the analysis results of computational provable security model is sound, the results of automatic evaluation framework is sound.

1.3 AP Designs

1.3.1 Cryptographic Hash Functions

Bellare, Canetti, and Krawczyk designed two authentication models in [1], the nested construction NMAC and the hash based scheme HMAC. The advantage of HMAC compared with MAC schemes constructed by block cipher is the fast processing speed and simpler in implementation. In this paper, the authors analyzed the security of NMAC and HAMC and provided a quantitative security result based on the security notion that HMAC is secure is the hash function used behaves like a PRF.

Suh et al. introduced an hash function design in [**?**] aiming to protect the integrity of message. Compare with early hash functions, what is their properties?

1.3.2 Message Authentication Code(MAC) Scheme

Iterated MAC Schemes For a deterministic MAC scheme, a secrete key is used in processing message M and generating the tag T. The deterministic MAC design works can be categroized to iterated MAC scheme and parallel MAC scheme.

In iterated MAC scheme, the process of a block in message M relys on the output of the process of its previous block. On the other hand, a block in message M can be processed only after its previous block finishes processing.

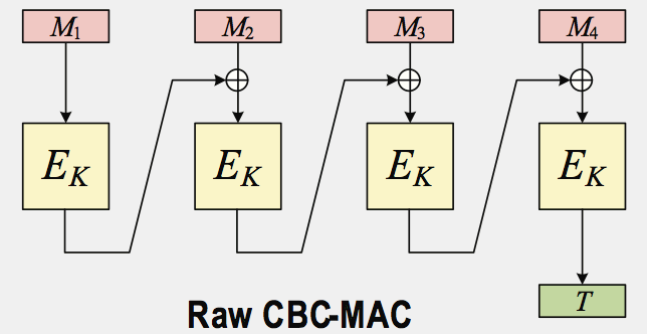

Figure 2: The Raw CBC-MAC

The research on iterated MAC schemes started from the raw CBC-MAC expressed in Figure 3.

Raw CBC-MAC The cryptographic primitive used in the raw CBC-MAC scheme is block cipher. The raw CBC-MAC is a secure MAC for only fixed length inputs. Assume the tag T of message M from CBC-MAC can be expressed as T = CBC-MAC$_k$(M), for another input M$\|$(M$\oplus$T), the tag T1=CBC-MAC(M$\|$(M$\oplus$T))=T. This attack indicates that the raw CBC-MAC is vulnerable if the input length is not fixed. Besides the vulnerability when the input length is not fixed, the raw CBC-MAC scheme suffers birthday attack, means the adversary needs only $2n/2$ input queries to succeed a forgery.

Bellare et al. provided the original security evaluation on raw CBC-MAC[3]. The conclusion is that if the adversary A is allowed to conduct arbitrary fixed length queries for q time, the probability that A succeeds in a forgery after the queries can be expressed with the queries times q and the computational assumption of block cipher used. Their quantitative conclusion showed that the forgery probability is very small.

EMAC The motivation of EMAC design is the problem that the raw CBC-MAC is secure only for fixed length inputs. EMAC is a optimized version of raw CBC-MAC providing security for arbitrary length inputs. The in the EMAC scheme, the final message block in the input is processed by a padding function to ensure that all the input blocks have the same length. Then raw CBC-MAC is applied applies and the result is encrypted by a the same block cipher in raw CBC-MAC with a different key.

Petrank and Rackof provided the original security evaluation of EMAC in [24]. The EMAC is secure for arbitrary length inputs under the security notion if the block cipher meets the related computational assumptions. However, the EMAC still suffers the birthday attack.

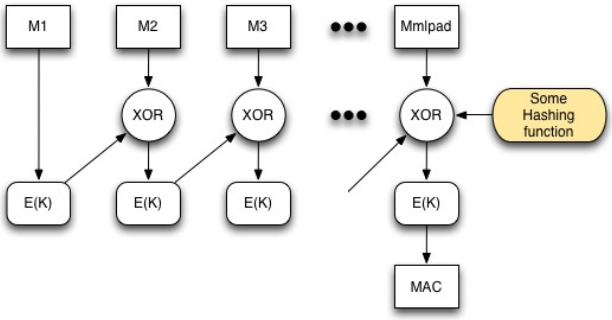

Figure 3: The class of CMAC

Hence the raw CBC-MAC is vulnerable if the input length is not fixed, a class

of MAC schemes named CMAC are developed to provide security for arbitrary length of inputs. The structure of the variants of CMAC is modeled in Figure 4.

XCBC: three key version of CMAC Black and Rogaway introduced a optimized version of EMAC called XCBC in [6]. There are two motivations of XCBC scheme: providing security for arbitrary length input; eliminate unnecessary padding operation on the input blocks in EMAC.

The XCBC scheme is a refined version of original EMAC. The contributions of XCBC scheme compared with the EMAC are: extending the domain to full domain; the block ciphers used in processing m input blocks is m other than m+1 in EMAC; all the block cipher in XCBC share a same key, the additional two keys are used to do exclusively or operation with the final input block,this design has faster processing speed.

Black and Rogaway provided a original security analysis based on computational approach of XCBC scheme in [6]. Their quantitative result was improved by Minematsu and Matsushima in [22]. Why the japanese want to optimize the original security bound? Any flaw in the proof? What is the advantage in the new proof procedure?

TMAC: two key version of CMAC Kurosawa and Iwata designed a optimized version of cmac requiring two different keys in [18] named TMAC. The two different keys used in processing the final input blocks in XCBC are replaced by a keyed hash function with two distinct constants as inputs. This optimization aims to reduce the key cost.

Kurosawa and Iwata provided an original security analysis on TMAC in [18] showing that the same level of security was achieved by TMAC compared with XCBC. The original quantitative security conclusion of TMAC was improved by Minematsu and Matsushima in [22].

OMAC: single key version of CMAC Iwata and Kurosawa introduced a optimization of original XCBC scheme with only one key, named OMAC in [13]. This key is used by the block cipher in the OMAC and the keyed hash function used in TMAC is replaced by a result of Galois field multiplication. OMAC is the variant of CMAC using the least number of keys.

Iwata and Kurosawa provided an original security analysis on OMAC in [13]. This result have no optimized version yet.

other iterated MAC designs Daemen and Rijmen introduced a iterated MAC design named ALRED-MAC in [8]. The motivation of of this work is the limitation of quantitative security result for iterated MAC scheme due to the birthday paradox.

Parallel MAC Schemes The motivation of designing MAC schemes in which input message blocks are processed in parallel is the latency of processing when

using iterated MAC schemes. This latency becomes more obvious when run the scheme on processors supporting pipeline of parallelism. In parallel MAC scheme, all the blocks in message M are processed in parallel. The output blocks of M processing are transformed to a single block B, then processed to generate tag T.

XOR MAC scheme Bellare, Guerin and Rogaway introduced a parallel MAC scheme named XOR MAC in [2]. The parallel input processing makes the xor MAC faster the CBC-MAC. In [2], the authors provided a quantitative security conclusion of xor mac that it is security under the security notion of message authentication system if the pseudorandom function used in the design is secure. The security bound of xor mac from [2] expressed a smaller result compared with the result of raw CBC-MAC.

The main weakness of xor mac is the cost of storage for additional input parameters. According the design of xor mac, assume the input message is divided into m blocks and the length of each block is half of the length of block cipher input. Each input block has a index with domain as [1,m]. A input block is the concatenated with the encoder of its index(whose length is same as the input block) to form the input to a block cipher, for example (i$\|$M[i]). Besides the m concatenated input blocks, a nonce formed with random number or counter is used a input to the block cipher. Then the m+1 output blocks from block cipher is xored and the output block of xor operation is concatenated with a random number or counter to form the final tag. We can see that the length of tag from xor-mac is the sum of the length of a output block from block cipher and the length of a random number or counter. This long-length MAC needs additional storage compared with the tag from iterated MAC schemes. On the other hand, the nonce maintained by the user of xor mac is regarded a disadvantage by some researchers.

Raw PMAC Black and Rogaway introduced an parallel MAC scheme named PMAC in [7]. The improvement of PMAC compared with XOR MAC is the less calling of block cipher when generating a tag. Secondly, there is no limitation for the input length when using PMAC. Another benefit brought by PMAC is that only one key is needed in generating the tag. Other processing operations include bit-level xor and gray code, which are cost-effective and fast processing.

The quantitative security result of PMAC was acquired by Black and Rogaway in [7]. They analysed the probability of internal collision in the arbitrary types of inputs and asserted that the PMAC design is a secure MAC if the block cipher used behaves like a pseudorandom permutation. In the proof from Black and Rogaway, the probability that an adversary can distinguish the PMAC from pseudorandom function is the probability that collsion of internal blocks(the outputs of block cipher) plus the probability of tag collision.

Lee et al. posted attacks on the raw PMAC in [19]. Their research indicated that the raw PMAC design did not bring advantage in security compared with iterated MACs. The raw PMAC suffers from their birthday like attack and the

key recover attack.

Tweakable block cipher and Optimized PMAC The concept of tweakable block cipher was formally defined by Liskov, Rivest, and Wagner in [20]. Rogaway introduced a efficient implementation of tweakable block cipher in [26]and adopt this implementation to replace the block cipher used in the original PMAC and OCB authenticated encryption mode. The application of tweakable block cipher in constructing PMAC can enhance the processing speed compared with original PMAC based on Grey code and simplify the structure of the scheme, which also simplifys the security analysis of the scheme .

In [26], Rogaway provided an assertion that the tweakable block cipher XE and XEX behave like a pseudorandom permutation if the block cipher inside is a pseudorandom permutation. Based on this assertion about security of tweakable block cipher, Rogaway gave an quantitative conclusion that the new PMAC based on tweakable block cipher is security under the security notion of message authentication system. This quantitative security result of tweakable block cipher based PMAC was provided in [26]. Minematsu and Matsushima optimzed this result in [22].

The evaluation procedure in [7] is questioned by the latter researchers hence the correlation between collision probabilty and distinguishing probability was not explained . This issue was also pointed in Nandi‘s improved analysis on PMAC in [23]. Nandi also provided a improved quantitative security result of PMAC which is smaller in all the cases than the one in [7] and [22].

iPMAC Sarkar introduce an optimisation on original PMAC in [30]. The galois field multiplication used in the tweakable block cipher introduced is replaced by a technique named tower field representation of Galois field. This replacement enhance the processing speed in software.

Hence the structure of iPMAC is same as the original PMAC except the input masking stage, the security evaluation by the author in [30] followed the approach in [7].

Stately MAC Schemes Deterministic MAC schemes cannot defend against the second type of attack in the threat model defined in Section 1.1.1 . The reason is that the secrete key used in genrating the tag will not be refreshed for each message. If two message are identical, their tags are identical. To address this issue, nonce is introduced to the tag generation in a MAC scheme. A nonce is a short message block whose value will be updated by each message in tag generation. To update the value of nonce, the address of message, a random number, a counter or the combination of these three elements are adopted in the nonce generation for each message. With the nonce introduced, a well designed stately MAC scheme should be capable in defending type 2 attack in threat model.

GMAC and GCM McGrew and Viega introduced the Galois/Counter Mode authenticated encryption design and the original security analysis is provided in [21]. The message authentication system in GCM(named GMAC) is a iterated scheme based on the concept of universal hashing. The block processing component used in GMAC is the Galois field multiplication other than the block cipher in deterministic MAC schemes. The motivation of GCM is to design a scheme combining counter mode of operation(CTR) with a message authentication code(the GMAC) to form a efficient and secure authenticated encryption system. One advantage of adopting the Galois field multiplication in GMAC is that it can be made easily in hardware and has efficient performance in software.

In the security analysis of GCM from McGrew and Viega, the GMAC was not discussed alone. The security of GMAC as a message authentication code is based on the fact that the encryption part in GCM is secure and the collision probability of ciphertext blocks is low. In GCM design, the nonce is called initialization vector (IV). For each invocation of GCM, the IV should be distinct.

In [14], Iwata, Ohashi and Minematsu provided an optimised analysis for the security of GCM. They pointed out the weakness in the lemma used in forming the bound of the quantitative result of security for GCM with a counter example. This counter example was developed to a distinguishing attack on GCM. Iwata et al. then provided a approach fixing the problem of original lemma and provided the new quantitative security result of GCM.

In [27], Rogaway pointed the weakness of GMAC as a MAC scheme when used alone that the security under adaptive chosen-message attack is not as good as the security of deterministic MAC schemes. On the other hand, GMAC requires the nonce for tag generation and this nonce should be maintained and refreshed by the user of GMAC. The potential issue of this design pointed by Rogaway is the reuse of nonce, which may lead to the collision of the tag.

Handschuh and Preneel introduced key-recovery attacks on universal hash function based MAC algorithms in [11]. They introduced two types of attacks: weak key finding and partial information leaks.

1.3.3 The Authenticated Encryption Schemes

Three format of AE Schemes: Encrypt-then-MAC, Encrypt-and-MAC, MAC-then-Encrypt. AREA is in the same format of MAC-then-Encrypt. Bellare After various kind of MAC schemes were evaluated and claimed to be secure, a new cryptosystem, the Authenticated Encryption(AE) was introduced The Authenticated to combine the encryption of plaintext blocks and generation of the MAC in a single scheme to provide both confidentiality and integrity. The methodology of constructing a AE scheme was categorized by Bellare in [4]. In this paper, the author claimed that according to the 3 types of modes defined, the most secure one was the Encrypt-then-MAC mode. In Rogawa‘s works [25] a systematical analysis about AE schemes using associate data was expressed.

Based the notion of security evaluation for AE systems introduced in [25], the security of several AE schemes in Encrypt-then-MAC mode have been analyzed systematically, including CCM [15]based on CBC-MAC,EAX[5] based on

OMAC, GCM [21] based on universal hashing(new bound revised in [14]), and OCB[28] based on PMAC(new bound revised in [20, 30]). Kasper and Schwabe introduced a implementation of AES that run faster than previous AES block cipher and adopted this implementation to construct a faster AES-GCM AE scheme[16].

There were research works on attacks to existing authentication system or AE systems, such as [29, 31, 17]. Some of these works recover the security weakness in the design works while some other works depict attacks that are out of the security bound of system analyzed.

1.4 Message Authentication(MA) Systems

What to say in this subsection

- The MA system design
- For each MA system, the security analysis and result

Concepts of Message Authentication(MA) System After the Authentication Primitives are evaluated systematically, researchers have paid effort on designing Message Authentication(MA) system to protect data integrity from or to the cache on-chip. In the design of Message Authentication(MA) system, the on-chip data is assumed to be inaccessible and secure, and the chip 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.

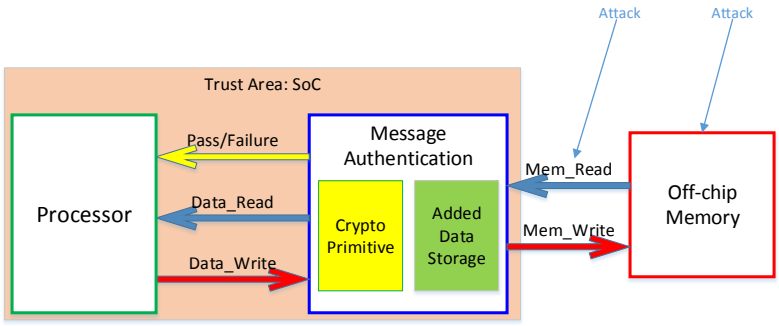

Figure 4: The Concept of MA System

Figure 1 expresses the concept of MA system for chip-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, add_B), where add_B represents an additional data block. The message block message(D,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,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 system finds that the D is tampered, MA sends a failure signal to processor, and invoke the exception handling procedure.

Security Evaluation for MA Systems As discussed with previous articles, encryption can only protect the integrity of data from and to the cache. Authentication Primitives(AP) should be adopted on the chip to process the data with additional information block, such as secret key or nonce.

The processed data will be transfered to the off-chip area in a specific message format. The message is ciphertext blocks for hash function, [ciphertext | tag] for MAC scheme, and E$_k$[plaintext | nonce] for AREA. The first step to evaluate the security of a MA system is figure out the AP system adopted and additional information needed.

For each kind of AP design, there are requirements on the inputs to the AP system. For example, a MAC scheme accepting a nonce requires the nonce to be distinct to each data that invoke the MAC scheme. This requirement ensure that the scheme can defend two types of attacks in the threat model. The MA system design based on such a MAC scheme should ensure that for each data write to the off-chip area from cache, the related nonce should be distinctive. Otherwise, the MA system design is not secure. The second step in security evaluation on a MA system is figure out whether the requirments on inputs of AP system is met.

If the MA system design meets the requirements of inputs for the AP system adopted, it is necessary to figure out what information is stored in off-chip area and what else are on-chip. Using MAC scheme as instance, the nonce and secrete key in tag generation should be inaccessable to the attacker. The third step to evaluation the security of an MA system is examine the information stored on-chip and off-chip. If there is some information that can be adopted by the attacker to break the authetication system, the MA design can be assume not secure.

The fourth step to evaluate the security of a MA system is analyzing the security of AP adopted. If the AP system is a novel design without previous systematically analysis, it is necessary to do the analysis with a evaluaiton mechanism, such as computational model based on provable security theory.

The final step is drawing the conclusion of the security of MA system. If the system does not fail in any one of previous four steps, the MA system is secure in general security notions and can effectively defend some kind of attacks.

Gassend et al. proposed an integrity protection architecture in 2003 [**?**]. The AP used in this design can be either hash function of MAC, not specificated.

Q: What is the contribution of this architecture? Off-chip processing faster? On-chip cost smaller? Can this architecture defend type 2 attack?

1.4.1 Hash Function Based MA Systems

When using hash function as AP to construct a MA system, the common solution is concatenation additional information to the data to form the input to hash function. This approach can ensure the distinction of input and protect against attacks in threat model.

Uniprocessor System Suh et al. introduced a architecture,named Log hash, to protect the integrity of data from and to the cache, published in 2003 [**?**]. In their design, the AP system adopted is a function named MSet-Add-Hash based on the hash function MD5. When cache writes a data to the off-chip area, the data is concatenated with the destination address and a timestamp to form a message chunk, which is sent the the MSet-Add-Hash and compute a hash value. The hash value is denoted as WriteHash and recored on-chip, and (data,timestamp) is writen on the off-chip area. When cache reads a data from memory, the (data,timestamp) is read to the chip. A ReadHash value is computed with the tuple (data,address,timstamp) and compared with the related WriteHash value. If these two hash values are identical, data is not tampered.

The security of Log hash design is achieved by concatenating address and timestamp to data in hash value generation. The concatenation ensure for each time of the invocation of hash function, the input is distinct. As MSet-Add-Hash is proved to be a secure hash function in [**?**], log hash design is secure to protect the data that invoke hash function.

One design weakness is that not all the data sent out and recevied to the chip will invoke the hash function on-chip. This issue exposes some data to the attacker. This weakness was pointed by Yan et al.[**?**] and Elbaz[**?**].

1.4.2 MAC Function Based MA Systems

Some MA systems are construcnted based on AE schemes to provide protection on both confidentiality and integrity of data. In these AE based MA system, the authentication part is constructed with MAC schemes.

Uniprocessor System XOM[**?**]: vulnerable to replay attack, analyzed by another paper. It‘s optimization [**?**] can defend replay attack.

In [**?**], Yan et al. proposed a MA design based on Galois/Counter Mode(GCM) authenticated encrypion. According to the original design article of GCM [**?**], the nonce used to encryption and tag generation for each data from cache should be distinct. Yan et al. introduced a split counter structure. Each data from cache is splited into sevel blocks,

- the AP is GCM, faster than hash(MD5)

- GCM require distinct IV as nonce for each invocation of GCM(each data)
- In Yan's Design, for each message , IV is distinct(counter split), How to ensure the IV distinct?
- the security of secret information used in authentication(KEY, iv) is protected by merkely tree, secure? why?
- assume the requirement is met, GCM is secure according to the analysis in [**?**], so the design is secure.
- What add data is stored on-chip? what is stored off-chip on memory? On-chip cost?

use timely authentication other than lazy authentication. use fast authentiation method other than hash(MD5).

Vaslin's design [**?**] adopted CRC checksum for authentication, CRC is less security compared to hash function and MAC scheme.

Rogers' design: [**?**] 2009:

- The AP is PMAC parallel processing
- to

Multiprocessor System Shi et al. proposed a MA system for multiprocessor shared memory architecture. It is an Encrypt-and-MAC scheme and the MAC is generated with hash function Sha 256.

References

[1] M Bellare, R Canetti, and H Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology - CRYPTO'96. 16th Annual International Cryptology Conference. Proceedings*, pages 1–15, Berlin, Germany, 1996.

[2] M Bellare, R Guerin, and P Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In *Advances in Cryptology - CRYPTO '95. 15th Annual International Cryptology Conference. Proceedings*, pages 15–28, Berlin, Germany, 1995.

[3] M Bellare, J Kilian, and P Rogaway. The security of cipher block chaining. In Yvo Desmedt, editor, *Advances in Cryptology Crypto 94*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. International Association for Cryptologic Research, Springer-Verlag, 1994.

[4] M Bellare and C Namprempre. Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In *Advances in Cryptology - ASIACRYPT 2000. 6th International Conference on the Theory and Application of Cryptology and Information Security. Proceedings*

(Lecture Notes in Computer Science Vol.1976), pages 531 – 45, Berlin, Germany, 2000.

[5] M Bellare, P Rogaway, and D Wagner. The EAX mode of operation. In *Fast Software Encryption. 11th International Workshop, FSE 2004. Revised Papers (Lecture Notes in Comput. Sci. Vol.3017)*, pages 389–407, 2004.

[6] J Black and P Rogaway. CBC MACs for arbitrary-length messages: the three-key constructions. In *Advances in Cryptology - CRYPTO 2000. 20th Annual International Cryptology Conference. Proceedings (Lecture Notes in Computer Science Vol.1880)*, pages 197–215, Berlin, Germany, 2000.

[7] J Black and P Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *Advances in Cryptology - EUROCRYPT 2002. International Conference on the Theory and Applications of Cryptographic Techniques. Proceedings (Lecture Notes in Computer Science Vol.2332)*, pages 384 —- 97, 2002.

[8] J Daemen and V Rijmen. A new MAC construction ALRED and a specific instance ALPHA-MAC. In *Fast Software Encryption. 12th International Workshop, FSE 2005. Revised Selected Papers (Lecture Notes in Computer Science Vol. 3557)*, pages 1–17, Berlin, Germany, 2005.

[9] O Goldreich, S Goldwasser, and S Micali. How to construct random functions. *J. Assoc. Comput. Mach. (USA)*, 33(4):792–807, 1986.

[10] S Goldwasser, S Micali, and R L Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[11] H Handschuh and B Preneel. Key-recovery attacks on universal hash function based MAC algorithms. In *Advances in Cryptology - CRYPTO 2008. 28th Annual International Cryptology Conference*, pages 144 —- 61, 2008.

[12] Mei Hong, Hui Guo, and Sharon X Hu. A cost-effective tag design for memory data authentication in embedded systems. In *Proceedings of the 2012 International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES 2012)*, pages 17–26, 2012.

[13] T Iwata and K Kurosawa. OMAC: one-key CBC MAC. In *Fast Software Encryption. 10th International Workshop, FSE 2003. Revised Papers (Lecture Notes in Comput. Sci. Vol.2887)*, pages 129 —- 53, 2003.

[14] T Iwata, K Ohashi, and K Minematsu. Breaking and Repairing GCM Security Proofs. In *32nd Annual Cryptology Conference. Advances in Cryptology - CRYPTO 2012*, pages 31–49, Berlin, Germany, 2012.

[15] J Jonsson. On the security of CTR+CBC-MAC. In *Selected Areas in Cryptography. 9th Annual International Workshop, SAC 2002. Revised Papers (Lecture Notes in Computer Science Vol.2595)*, pages 76–93, 2003.

[16] E Kasper and P Schwabe. Faster and timing-attack resistant AES-GCM. In *Cryptographic Hardware and Embedded Systems - CHES 2009. Proceedings 11th International Workshop*, pages 1–17, 2009.

[17] Markus G Kuhn. Cipher instruction search attack on the bus-encryption security microcontroller DS5002FP. *IEEE Transactions on Computers*, 47(10):1153–1157, 1998.

[18] K Kurosawa and T Iwata. TMAC: two-key CBC MAC. In *Topics in Cryptology - CT-RSA 2003. Cryptoghraphers' Track at the RSA Conference 2003. Proceedings (Lecture Notes in Computer Science Vol.2612)*, pages 33–49, Berlin, Germany, 2003.

[19] Changhoon Lee, Jongsung Kim, Jaechul Sung, Seokhie Hong, and Sangjin Lee. Forgery and key recovery attacks on PMAC and Mitchell's TMAC variant. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4058 LNCS, pages 421–431, Melbourne, Australia, 2006.

[20] M. Liskov, R.L. Rivest, and D. Wagner. Tweakable block ciphers. pages 31 – 46, Berlin, Germany, 2002//. tweakable block ciphers;cryptographic primitive;cryptographic key;initialization vector;CBC mode;OCB mode;primitive block-cipher level;security;tweak block chaining;.

[21] D A McGrew and J Viega. The security and performance of the Galois/counter mode (GCM) of operation. In *Progress in Cryptology - INDOCRYPT 2004. 5th International Conference on Cryptology in India. Proceedings (Lecture Notes in Computer Science Vol.3348)*, pages 343 – 55, Berlin, Germany, 2004.

[22] K Minematsu and T Matsushima. New bounds for PMAC, TMAC, and XCBC. In *Fast Software Encryption. 14th International Workshop, FSE 2007. Revised Selected Papers. (Lecture Notes in Computer Science vol. 4593)*, pages 434 —- 51, 2007.

[23] M Nandi and A Mandal. Improved security analysis of PMAC. *J. Math. Cryptol. (Germany)*, 2(2):149 – 62, 2008.

[24] F Petrank and C Rackoff. CBC MAC for real-time data sources. *J. Cryptol. (USA)*, 13(3):315 – 38, 2000.

[25] P Rogaway. Authenticated-encryption with associated-data. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 98–107, Washington, DC, United states, 2002.

[26] P Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In *Advances in Cryptology-ASIACRYPT 2004. 10th International Conference on the Theory and Application of Cryptology and Information Security. Proceedings (Lecture Notes in Computer Science Vol.3329)*, pages 16–31, Berlin, Germany, 2004.

[27] P Rogaway. Evaluation of some blockcipher modes of operation. 2011.

[28] P Rogaway and J Black. OCB: a block-cipher mode of operation for efficient authenticated encryption. *ACM Trans. Inf. Syst. Secur. (USA)*, 6(3):365–403, 2003.

[29] M-JO Saarinen. Cycling attacks on GCM, GHASH and other polynomial MACs and hashes. In *Fast Software Encryption. 19th International Workshop, FSE 2012. Revised Selected Papers*, pages 216 —– 25, 2012.

[30] P Sarkar. Pseudo-random functions and parallelizable modes of operations of a block cipher. *IEEE Transactions on Information Theory*, 56(8):4025–4037, 2010.

[31] Z Yuan, W Wang, K Jia, G Xu, and X W. New birthday attacks on some MACs based on block ciphers. In *Advances in Cryptology - CRYPTO 2009. Proceedings 29th Annual International Cryptology Conference*, pages 209 – 30, Berlin, Germany, 2009.