Asian Journal of Information Technology, 5 (3) : 298-305, 2006

The Tag Secrecy of Authentication and its Application to Implementing Secure Channels

Zhenyu Hu, Dongdai Lin and Wenling Wu
State Key Laboratory of Information Security, Institute of Software of Chinese Academy of Sciences, Beijing, People's Republic of China, 100080
Graduate School of the Chinese Academy of Sciences, Beijing, People's Republic China, 100039

Abstract: Message authentication, the most important equipment for access control, is a technique to prevent a message from being invalid modified during transmission across Internet. Its security involves the unforgeability only. However, in the access control practice it is usually combined with data encryption to implement secure channels, in which we would like the authentication tag does not disclose the message information (e.g., SSH). We refer to this problem as the tag secrecy of authentication and deal with it in this paper. We present a new security notation, tIND-CMA (tag-indistinguishability against Chosen-Message Attacks), to characterize the tag secrecy of an authentication scheme, discuss its application to implementing secure channels. The results show that, for some common modes of encryption, CBC (cipher-block chaining) and OTP (one time pad) or CTR(counter), a tIND-CMA secure authentication scheme enables the MAC-and-Encrypt to be otherwise insecure to implement secure channels.

Key words: Access control, authenticity, Message Authentication Code (MAC), privacy, secure shell (SSH), secure channel

INTRODUCTION

Access control is the main means to guarantee that services and information are providedto the intended parties. The usually way for a party to access the Internet recourses (e.g., Web services) is that the purport provides some identity information along with his pass-code to the verifier. Is it possible that an adversary modifies the identify information and impersonates a valid user to access the restricted recourse? The answer is yes. To prevent the message from being modified during transmitting across a network, the authenticity of data is required,and authentication techniques are developed accordingly.

The authenticity security of data requires that the receiver have confidence that the communicated data does originate with the claimed sender. The authentication problem is very different from the encryption problem. We are not worried about secrecy of the data; instead, we are worried about the adversary modifying it.

Apart from the authenticity, the secrecy of data is always required for access control. When a user provides his identity information and pass-code to verify, he would hope that this information would not be disclosed to a third party, even the verifier. To this end, encrypt techniques are usually incorporated with authentication technique. Secure Shell (SSH) protocol[1,2], which has become one of the most popular and widely used cryptographic protocols on the Internet, is a typical mode of this combination.

The cryptographic heart of the SSH protocol is its Binary Packet Protocol (BPP)[1,2], which, working in the MAC-and-Encrypt manner, is responsible for the underlying symmetric encryption and authentication (or the authenticated encryption) of all messages sent between two parties involved in an SSH connection. Although the current version of SSH (implemented with HMAC) is secure, the MAC-and-Encrypt paradigm is not generally secure[3], because that the underlying authentication scheme may disclose the valuable information about the plaintext in its authentication tags (message authentication code). To guarantee the combination of MAC-and-Encrypt a secure encryption scheme, we would like, in addition to the unforgeability, the authentication tag does not disclose the message information. We can regard the tagging algorithm of an authentication scheme as mix-transformaton, too and hope that the authentication tag can mask the information of the authenticated message. We refer to this privacy as tag secrecy of an authentication scheme, characterize it and discuss its application in this study.

Theoretically, the previous security notions for authentication schemes address unforgeability only. In

Corresponding Author: Zhenyu Hu, State Key Laboratory of Information Security, Institute of Software of Chinese Academy of Sciences, Beijing, People's Republic of China, 100080

this study, we inspect the other side of authentication technique. We would like that the authentication tag of an authentication scheme reveals no information about the message. Though many of the authentication schemes are constructed from cipher blocks or secure hash functions and are pseudorandom, there are authenticated schemes that are secure in the sense of unforgeability but can disclose the message information in their tags (Theorem 3). So we think studying the tag secrecy of an authentication scheme is not trivial, particularly in the application to implementing secure channels with the mode of MAC-and-Encrypt. We formally discuss this problem, extend the security notion of privacy into authentication schemes and propose a new notion of tag secrecy-tIND-CMA (tag-indistinguishability against Chosen-Message Attacks) and its cousin tIND-DCMA (tag-indistinguishability against Distinct Chosen-Message Attacks).

With regard to the rationality of tag secrecy of MAC scheme, we illustrate some popular authentication schemes that satisfy the tag secrecy of tIND-CMA or tIND-DCMA. We discuss the relationship between tIND-CMA and the pseudorandom function, showing that the existence of pseudorandom function is sufficient for the tIND-CMA security.

We also discuss the application of tag secrecy of MAC. Based on the notions of tag secrecy, we show how secure the MAC-and-Encrypt is. Particularly, we show that tIND-DCMA security of MAC, for some common modes of encryption, CBC (with a secure underlying block cipher) and OTP (stream ciphers that xor data with a pseudorandom pad) or CTR, enables the MAC-and-Encrypt paradigm to implement secure channels.

We discuss the notion of tag secrecy. First, we propose the definition of tIND-CMA security and its cousin tIND-DCMA security, showing the relationship between the tag secrecy and the pseudorandom function. Next, we study the tag secrecy of some popular MAC scheme including the HMAC and CBC MAC. In addition, we discuss the problem of convert a MAC scheme into a tIND-CMA secure MAC while preserving the unforgeability of the original scheme. We show the applications of tag secrecy to guaranteeing the security MAC-and-Encrypt and implementing secure channels.

PRELIMINARIES

Conventions: Suppose x and y are strings, let x||y denote the concatenation of x and y. When we say an algorithm is stateful, we mean that it uses and updates its state and that it maintains the state between invocations. If f is a randomized (resp., deterministic) algorithm, then $x \xleftarrow{R} f(y)\left(\text{resp.}, x \leftarrow f(y)\right)$ denotes the process of running f on input y and assigning the result to x. If A is a program or algorithm, A⇐x means return the value x to A. If $s \geq 1$ and i are integers with $0 \leq i \leq 2^s-1$ then we let $NtS_s(i)$(read number to string) denote the s-bit string which is the binary representation of integer I.

Message authentication schemes and their security: A message authentication scheme MA = (K, T, V) consists of three algorithms. The randomized key generation algorithm K takes input a security parameter k∈N and returns a key K; We write $K \xleftarrow{R} K(k)$. The tagging algorithm T could be either randomized or stateful.It takes the key K and a message M to return a tag t; we write $t \xleftarrow{R} T_K(M)$. The verification algorithm V is deterministic. It takes the key K, a message M and a candidate tag t for M to return a bit v; we write $v \leftarrow V_K(M, t)$. We require that VK(M, TK(M)) = 1 for all M∈{0, 1}*. The scheme is said to be deterministic if the tagging algorithm is deterministic and verification is done via tag re-computation. We sometimes call a message authentication scheme a MAC and also sometimes call the tag t a MAC.

Usually, the security for message authentication concerns the unforgeability, which considers an adversary F who is allowed a chosen-message attack, modeled by allowing it access to an oracle for TK(·). F is successful if it can make the verifying oracle VK(·, ·) accept a pair (M, t) that was not legitimately produced. There are two measures with regard to what legitimately produced can mean. The standard measure is that the message M is new, meaning F never made query M of its tagging oracle. This type of forgery is called weak forgery. A more stringent measure considers the adversary successful even if the message is not new, as long as the tag is new. This type of strong forgery means that the adversary wins as long as t was never returned by the tagging oracle in response to query M. In the formal definition, the notations WUF-CMA and SUF-CMA are used respectively for weak and strong unforgeability against chosen-message attacks.

Definition 1: (Security of Message Authentication Scheme[4]) Let MA = (K, T, V) be a message authentication scheme. Let k∈N and let Fw and Fs be adversaries that haveaccess to two oracles. Consider the following experiment:

Experiment $Exp^{wuf-cma}_{MA,F_w}(k)$

$K \xleftarrow{R} K(k)$
If F_w makes a query M' to the oracle $T_k(\cdot)$
Then
$t' \xleftarrow{R} T_k(M')$; $F_w \Leftarrow t'$
If F_w makes a query (M,t)
to the oracle $V_k(\cdot,\cdot)$ such that
$V_k(M, t)$ returns 1 and
M was never queried to
the oracle $T_k(\cdot)$,
then return 1 else retun 0.

Experiment $Exp^{suf-cma}_{MA,F_s}(k)$

$K \xleftarrow{R} K(k)$
If F_s makes a query M' to the oracle $T_k(\cdot)$
Then
$t' \xleftarrow{R} T_k(M')$; $F_s \Leftarrow t'$
If F_s makes a query (M, t)
to the oracle $V_k(\cdot,\cdot)$ such that
$V_k(M, t)$ returns 1 and
t was never returned by the
oracle $T_k(\cdot)$, in response to query M.
then return 1 else retun 0.

The advantages of the forgers are defined via

$$Adv^{Wuf-cma}_{MA,F_w}(k) = Pr[Exp^{wuf-cma}_{MA,F_w}(k)=1]$$

$$Adv^{suf-cma}_{MA,F_s}(k) = Pr[Exp^{suf-cma}_{MA,F_s}(k)=1]$$

The scheme MA is said to be WUF-CMA secure (resp. SUF-CMA secure) if the function

$$Adv^{Wuf-cma}_{MA,F}(k) (resp.\ \ Adv^{suf-cma}_{MA,F}(k))$$

is negligible for any forger F whose time complexity is polynomial in k.

Secure channels: Secure channel was first introduced by Canetti and Krawczyk[5] to model secure communications which is intended to capture the standard network-security practice in which communications over public networks are protected through sessions between pairs of communicating parties and where each session consists of two stages. First, the two parties run a key-exchange protocol that establishes an authenticated and secret session key shared between the parties. Then, in the second stage, this session key is used, together with symmetric-key cryptographic functions, to protect the integrity and/or secrecy of the transmitted data. The formalism of[5] involves the definition of a key-exchange protocol for implementation of the session and key establishment stage, as well as of two functions, SND and RCV, that define the actions applied to transmitted data for protection over otherwiseinsecure links. Specifically, when a message M is to be transmitted from party S to party R under a session established between two parties, the function SND is applied to M and, possibly, to additional information such as a message identifier. The definition of SND typically consists of the application of some combination of a MAC and symmetric encryption keyed via the session key. The function RCV describes the action at the receiving end R for decoding and verifying incoming messages and it typically involves the verification of a MAC and/or the decryption of an incoming ciphertext. A protocol that follows this formalism is called in[5] a network channels protocol and its security is defined in terms of authentication and secrecy.

In study[5], authentication is achieved by the protocol if any message decoded and accepted as valid by the receiving party to a session was indeed sent by the partner to that session. (That is, any modification of messages produced by the attacker over the communications links, including the injection or replay of messages, should be detected and rejected by the recipient; in[5] this is formalized as the emulation of an ideally-authenticated channel.) Secrecy is formalized in the tradition of semantic security: among the many messages exchanged in a session the attacker chooses a pair of test messages of which only one is sent; the attacker's goal is to guess which one was sent. Security is obtained if the attacker cannot guess correctly with probability significantly greater than 1/2. A network channels protocol is called a secure channels protocol if it achieves both authentication and secrecy in the sense outlined above.

In this study, we are not concerned here with a specific key-exchange mechanism, but rather assume a secure key-exchange protocol[5] and even an ideally shared session key. In stead, we focus on authenticated encryption schemes that are used as specific SND and RCV functions. We say that any authenticated encryption scheme implements secure channels if when used as the specification of the SND and RCV functions the resultant protocol is a secure channels protocol.

TAG SECRECY OF MESSAGE AUTHENTICATION SCHEMES

In this section we formally discuss the tag secrecy of authentication schemes, present a new notation to characterize it. We take some examples to illustrate the existence of the new security and produce a simple method to convert a MAC scheme that is insecure in the

new sense of security to a tag secrecy MAC scheme, while preserves its unforgeability.

Tag secrecy of message authentication schemes: Following the notion of privacy of an encryption scheme, we consider the indistinguishibility of tags of a MAC scheme. We consider an adversary who can choose arbitrary two messages of the same length M_0, M_1, then queries the left-or-right tagging oracle $T_K(LR(\cdot,\cdot,b))$ of them. A bit b is chosen under the rug(not known to the adversary) and the oracle tags one of the messages according to the value of b. if b = 0, the oracle returns $T_K(M_0)$ to the adversary, otherwise, returns $T_K(M_1)$. The goal of the adversary is to guess the correct value of the bit b. we consider an authentication scheme to be tIND-CMA (tag-indistinguishability against Chosen-Message Attacks)secure, if a reasonable adversary cannot obtain significant advantage in distinguishing the case b = 0 and b = 1, given access to the racle. Formal description is as follows:

Definition 2: (tIND-CMA security of MAC Scheme) Let MA = (K, T, V) be a message authentication scheme. Let $b\in\{0,1\}$. Let A be an adversary that has access to an oracle TK(LR(·,·,b)). Consider the following experiment:

$$\text{Experiment}\quad \text{Exp}_{MA}^{\text{tind-cma-b}}(A_{cma})$$
$$K \xleftarrow{R} k$$
$$\text{Run}\quad A_{cma}^{Tk(LR(\bullet,\bullet,b))}$$

Replies to $T_K(LR(M_0, M_1, b))$ queries as follows:
$t \leftarrow TK(Mb)$; $A_{cma} \Leftarrow t$
Until A_{cma} returns a bit b'
Return b'
We define the advantage of A via:

$$\text{Adv}_{MA}^{\text{tind-cma}}(A) = \text{pr}[\text{Exp}_{MA}^{\text{tind-cma-1}}(A)=1] - \text{Pr}[\text{Exp}_{MA}^{\text{tind-cma-0}}(A)=1]$$

The authentication scheme MA is tIND-CMA secure if, for any adversary A, the advantage $\text{Adv}_{MA}^{\text{tind-cma}}(A)$ is negligible.

If we mandate in the Definition 2 that all left messages of A's queries be unique and that all right messages of A's queries be unique, then we get the notion of tIND-DCMA (tag-indistinguishability against Distinct Chosen-Message Attacks), which is similar to the notion IND-DCPA in[6].

Recall that, in the actual network communication, each message is assigned a unique identifier (sequence number), which guarantees that all the input of tagging oracle are always distinct from each other. We refer to this mode as MAC-CTR (counter) and formally describe as follows:

Definition 3: (MAC-CTR) Let MA = (K, T, V) be a message authentication scheme, ctr be a counter of length of s. The MAC-CTR scheme using MA, denoted by work as follows:

Agorithm $\overline{K}$	**Algorithm $\overline{T}_k(M)$**	**Algorithm $\overline{V}_k(M,\sigma)$**
$K \xleftarrow{R} k$	**If ctr $\geq 2^{s-1}$ return $\perp$**	**Parse σ as c‖t**
ctr $\leftarrow$ 0	**$t \leftarrow T_k(NtS_s(ctr)\|M)$**	**$v \leftarrow V_k(c\|M,t)$**
Return K	**$\sigma \leftarrow NtS_s(ctr)\|t$**	**return v**
	ctr $\leftarrow$ ctr+1	
	return σ	

MA – CTR = $(\overline{K}, \overline{T}, \overline{V})$,
Obviously, the following theorem holds:

Theorem 1: (Relation between tIND-CMA and tIND-DCMA) An authentication scheme is tIND-DCMA secure if and only if it is tIND-CMA in the MAC-CTR mode.

In point of actual network communication, the tIND-CMA and tIND-DCMA are equivalent, so we sometimes call both of them tag secrecy indiscriminately.

The intuitive idea of the tIND-DCMA security comes from the mixing-transformation property of an encryption algorithm, where the semantic security essentially means that a cipher text has a distribution in the message space that is indistinguishable from the uniform distribution in the same pace; in other words, the encryption algorithm is a pseudorandom function. However, a pseudorandom function would be a deterministic algorithm, which means that it will output two identical strings if it is fed the same message twice, which leads that the IND-CPA security of a stateless and deterministic algorithm can be easily broken. The same problem exists in discussing the tag secrecy of a MAC scheme.

The relation between the tIND-DCMA security and the pseudo-randomness can be described as the following theorem. We refer the proof to[7]:

Theorem 2: (Relation between Tag-secrecy and PRF) A PRF function is also a tIND-DCMA secure function. More formally, let F be a PRF function, for any IND-DCMA-adversary A attacking F. we can design a PRF distinguisher D such that

$$\text{Adv}_F^{\text{tind-dcma}}(A) \leq 2 \cdot \text{Adv}_F^{\text{prf}}(D) \qquad (1)$$

Furthermore, D will use the same resource as A.

Why we need to discuss the tag secrecy of authentication schemes? Usually, the security of authentication schemes is related to the unforgeability only. The unforgeability of an authentication scheme may not guarantee its tag secrecy. In fact, for any secure

authentication scheme, we can construct a security authentication scheme that can immediately disclose the plaintext information in its tag, while preserve the unforgeability.

Theorem 3: (Unforgeability does not imply Tag-secrecy) Given any WUF-CMA (resp. SUF-CMA) secure MAC scheme MA = (K, T, V). We can construct a message authentication scheme MA' such that MA' is WUF-CMA (resp. SUF-CMA) secure, but is neither tIND-CMA nor tIND-DCMA secure.

Proof of Theorem 3: Let MA = (K, T, V) be the given MAC scheme, we construct a MAC scheme MA' which is the same as the given one except that it pre-appends the first bit of the original message to the tag. Formally MA' = (K, T', V') has the same key generation algorithm as the given MAC scheme and the following tagging and verification algorithms:

Algorithm $T'_k(M)$	Algorithm $V'_k(M, t)$
Parse M as x‖M' where x is a bit	Parse M as x‖M' where s is a bit
Return x‖$T_k(M)$	Parse t as s‖t' where s is a bit
	If x = s and $V_k(M,t') = 1$
	Then return 1 else return 0

It is easy to see that if MA is WUF-CMA (resp. SUF-CMA) secure then MA' is WUF-CMA (resp. SUF-CMA) secure. However, the MA' fails to achieve either tIND-CMA security or tIND-DCMA security because the first bit of the message is provided to the adversary via the MAC tag.

Examples of tag secrecy MAC: Let us take some examples to illustrate the common existence of tag secrecy of authentication schemes.

Example 1: (Pseudorandom function MAC): A general method for designing MACs is to make use of the fact that any pseudorandom function is in fact a MAC[8]. On the other hand, from the Theorem 2, its mixing-transformation property makes it the ideal tIND-DCMA secure MAC.

Example 2: (CBC MAC[8]): Let F: $\{0,1\}^k \times \{0,1\}^l \{0,1\}^l$ be a block cipher. The CBC MAC with base family F is the (deterministic) message authentication scheme in which the tag of a message is the last block of ciphertext obtained by processing the message in CBC mode with zero IV. More details are as follows:

Algorithm $T_k(M)$	Algorithm $V_k(M, \sigma)$
Divide M into l bit blocks,	Divide M into l bit blocks,
$M = x_1 \ldots x_n$ $y_0 \leftarrow 0^l$	$M = x_1 \ldots x_n$ $y_0 \leftarrow 0^l$
For i = 1, ..., n do $y_i \xleftarrow{R} F_k(y_{i-1} \oplus x_i)$	For i = 1, ..., n do $y_i \xleftarrow{R} F_k(y_{i-1} \oplus x_i)$
Return Y_n	If $Y_n = \sigma$ then return 1 else return 0_n

Where FK denotes the specific function of F under the chosen key K. The message M input to the algorithm above must have length a multiple of l bits.

Given a finite PRF family, the CBC transformation preserves pseudorandomness. According the security of CBC against chosen-plaintext attack[9], the CBC MAC is tIND-DCMA secure but is not tIND-CMA secure (stateless and deterministic). However, if the initial vector IV is chosen randomly rather than 0, then the CBC MAC is tIND-CMA secure.

Example 3: (HMAC[10,11,12]): Let H be a cryptographic hash function, B the byte-length of hashing block and L the byte-length of hash outputs. Suppose two fixed and different strings ipad and opad as follows (the 'i' and 'o' are mnemonics for inner and outer):

ipad = the byte 0×36 repeated B times
opad = the byte 0×5C repeated B times.

The function HMAC takes the key K and 'text' and produces

$$HMACK(text) = H(K \oplus opad, H(K \oplus ipad, text))$$

It had been shown that if the underlying hash is collision-resistance and randomness, then the HMAC is unforgeable[11]. In relation to the tag secrecy of HMAC, it's easy to see that, if the underlying hash is pseudorandom, the output of it meet the notion of tIND-DCMA security.

In the SSH, all of the recommended MAC schemes are HMAC[6]. The 'sequence-number' used during computing the MAC tags turns the HMAC into the MAC-CTR, enabling the authentication scheme used in SSH tIND-CMA secure.

Building a tag secrecy MAC out of a secure MAC: Initially, the MAC is intended for integrity rather than privacy, so a secure MAC scheme may not be tag secrecy scheme under the condition of chosen-message attacks. In this section we discuss how to build a tag secrecy MAC out of a secure MAC scheme (i.e., unforgeable against chosen message attack) that is not tag secrecy, while preserves the unforgeability.

The idea here is simple. To get a MAC tag, we first transform the original message into a random string, then tag the intermediate random string and get the final MAC tag. By the pseudorandom property of transformation, we mask the original information and make the final tags secrecy. An ideal pseudorandom transformation is cryptographic hashing.

Definition 4: (Hash-then-MAC) Let H = (Kh, Hash), MA = (Kt, T, V) be keyed hashing and message authentication schemes with compatible message spaces (the outputs from Hash are suitable inputs to T); let s be a parameter of appropriate size(e.g. the size of security parameter of the base encryption scheme). We construct a composite hash-then-MAC scheme $\overline{MA} = \overline{K}, \overline{T}, \overline{V}$ as follows:

Algorithm $\overline{K}$	Algorithm $\overline{T}_{<K_h,K_t>}(M)$	Algorithm $\overline{V}_{<K_h,K_t>}(\overline{M},\sigma)$
$K_h \xleftarrow{R} k_h$; $K_t \xleftarrow{R} k_t$	$h \leftarrow Hash_{K_h}(M)$	$h \leftarrow Hash_{K_h}(M)$
Return $<K_h,K_t>$	$\sigma \leftarrow T_{K_t}(h)$	$v \leftarrow V_{K_t}(h, \sigma)$
	Return	Return v

Comparing the above Hash-then-MAC scheme to the normal MAC scheme, Hash-then-MAC executes an extra hashing operation both before the tagging and verification operation, which would degrade the performance of the scheme. However, noticing the data compressing property of the hash scheme and the fact that hashing operation is usually much fast, the hashing operation does not obviously spoil the entire performance, especially when the original messages are long.

The following theorems state the unforgeability and tag secrecy of the composed Hash-then-MAC scheme via Definition 4. It is easy to prove them and we omit it for conciseness.

Theorem 4: (Unforgeability of Hash-then-MAC) Let MA be a message authentication scheme and H be a keyed hashing scheme. Let $\overline{MA}$ be the Hash-then-MAC scheme associated to them as per Definition 4. If the hash scheme is pseudorandom, then $\overline{MA}$ preserves the unforgeability of the underlying MAC. Concretely, given any UF-CMA adversary A against $\overline{MA}$, we can construct adversaries F and C such that

$$\mathrm{Adv}_{\overline{MA}}^{\mathrm{wuf-cma}}(A) \leq \mathrm{Adv}_{MA}^{\mathrm{wuf-cma}}(F) + \mathrm{Adv}_{H}^{\mathrm{coll-cma}}(C) \qquad (2)$$

$$\mathrm{Adv}_{\overline{MA}}^{\mathrm{suf-cma}}(A) \leq \mathrm{Adv}_{MA}^{\mathrm{suf-cma}}(F) + \mathrm{Adv}_{H}^{\mathrm{coll-cma}}(C) \qquad (3)$$

Furthermore, F and C use the same resources as A except that F's messages to its tagging and tag verification oracles may be shorter than A's tagging queries (due to the hashing).

Theorem 5: (Tag secrecy of Hash-then-MAC): Let MA, H be a message authentication scheme and a keyed hashing scheme respectively. Let $\overline{MA}$ be the MAC-and-Encrypt scheme associated to them as per **Definition 4:** Then, if H is pseudorandom function, then $\overline{MA}$ is tIND-DCMA secure. Concretely, given any adversary A against tIND-DCMA, we can construct a PRF distinguisher D, such that,

$$\mathrm{Adv}_{\overline{MA}}^{\mathrm{tind-dcma}}(A) \leq 2 \cdot \mathrm{Adv}_{H}^{\mathrm{prf}}(D)$$

APPLICATION OF TAG SECRECY MAC

In this section we discuss how the tag secrecy MAC is applied to security concerned affairs. The main application of tag secrecy MAC is to compose authenticated encryption scheme by the MAC-and-Encrypt method to implement secure channel.

The construction of MAC-and-Encrypt: Notice that, in the real network setting, each of messages transmitted across the Internet is assigned a unique identifier (sequence number). To suit to this situation, we define the composite MAC-and-Encrypt as follows:

Definition 5: (MAC-and-Encrypt) Let SE = (Ke, E, D), MA = (Km, T, V) be an encryption and a message authentication scheme with same message spaces. Let ctr be a counter of length of s. The MAC-and-Encrypt scheme $\overline{SE} = \overline{K}, \overline{E}, \overline{D}$ is defined as follows:

In the rest of this paper, we sometimes use the note M&E to denote the MAC-and-Encrypt.

The security of MAC-and-Encrypt: Generally, the MAC-and-Encrypt paradigm does not preserve privacy because the MAC could reveal information about the plaintext, although it inherits the integrity of the MAC in a direct way and preserves integrity of plaintexts. However, in the case of tag secrecy of MAC, the MAC-and-Encrypt paradigm does preserve the IND-CPA security of the underlying encryption scheme. Particularly, for some common modes of encryption, CBC (with a secure underlying block cipher) and OTP (stream ciphers that xor data with a pseudorandom pad) or CTR, the MAC-and-Encrypt does work for implementing secure channels.

First of all, in the similar way of[3], we derive a theorem that can be proven in the security model of[5]

Theorem 6: (Derived from[5]) Let SE be an IND-CPA encryption function and MA a MAC function. If the composed scheme MAC-and-Encrypt, considered as an encryption scheme, is IND-CPA secure and INT-CTXT secure, then MAC-and-Encrypt implements secure channels.

For the sake of completeness, we present the plaintext integrity of MAC-and-Encrypt paradigm without proof.

Theorem 7: (Integrity of MAC-and-Encryp[4]) Let SE be a symmetric encryption scheme, let MA be a message authentication scheme and let $\overline{SE}$ be the encryption schemeobtained from SE and MA via the Definition 5. If MA is WUF-CMA-secure, then $\overline{SE}$ is INT-PTXT secure. Concretely, for any adversary

$$Adv_{EJM}^{int-ptxt}(F) \leq Adv_{MA}^{wuf-cma}(F) \quad (5)$$

The next theorem tells that MAC-and-Encrypt with a tag secrecy MAC does preserve the security of the underlying encryption scheme.

Theorem 8: (Privacy of MAC-and-Encrypt) Let SE be a symmetric encryption scheme, let MA be a message authentication scheme and let $\overline{SE}$ be the encryption scheme obtained from SE and MA via the Definition 5. If MA is tIND-CMA-secure and SE is IND-CPA secure, then $\overline{SE}$ is IND-CPA secure. Concretely, for any adversary A

$$Adv_{\overline{SE}}^{ind-cpa}(A) \leq Adv_{SE}^{ind-cpa}(A) + Adv_{MA}^{tind-cma}(A) \quad (6)$$

Proof of Theorem 8: Let A denote an IND-CPA adversary against $\overline{SE}$ that has oracle access to $\overline{E}_K$ (LR(•, •, b)), b∈{0,1}. Let x∈{1, 2, 3}. We define three experiments associated with A as follows.

Experiment $ExpG_x$

$$K_e \xleftarrow{R} k_e; K_t \xleftarrow{R} k_t$$

Run A replying to its oracle query (M0, M1) as follows:
Switch (x):

ctr ←ctr+1; $M'_0 \leftarrow NtS(ctr)||M_0$; $M'_1 \leftarrow NtS(ctr)||M_1$
Case x = 1: σ ←TKt(M'_1); c'←EKe(M'_1)
Case x = 2: σ ←TKt(M'_1); c'←EKe(M'_0)
Case x = 3: σ ←TKt(M'_0); c'←EKe(M'_0)
A ⇐'||σ
Until A halts and returns a bit b'
Return b'.

Let Px = Pr [x ExpG = 1] denote the probability that experiment x ExpG (x∈{1, 2, 3}) returns 1, By the definition of $Adv_{\overline{SE}}^{ind-cpa}(A)$, we have

$$Adv_{\overline{SE}}^{ind-cpa}(A) = P_1 - P_3 = (P_1 - P_2) + (P_2 - P_3) \quad (7)$$

It's easy to see that $ExpG_1$, combining with $ExpG_2$, simulates the attack experiment against the IND-CPA-security of SE and, $ExpG_3$, combining with $ExpG_2$, simulates the attack experiment against the tIND-CMA-security of MA. Thus:

$$P_1 - P_2 \leq Adv_{SE}^{ind-cpa}(A) \quad (8)$$

$$P_2 - P_3 \leq Adv_{MA}^{ind-cpa}(A) \quad (9)$$

Combining equation(7)~(9), so

$$Adv_{\overline{SE}}^{ind-cpa}(A) \leq Adv_{SE}^{ind-cpa}(A) + Adv_{MA}^{timd-cma}(A)$$

and Theorem 8 holds.

The Theorem 7 and Theorem 8 show that combining with the IND-CPA security of the underlying encryption scheme, the tag secrecy of MAC provides the INT-PTXT and IND-CPA security for the MAC-and-Encrypt paradigm. However, under this assumption, the integrity of ciphertext or the IND-CCA2 security may not be guaranteed. But if the underlying encryption scheme is a pseudorandom permutation, then any modification on the ciphertext will be reflected to the plaintext, which can be easily detected by the underlying MAC. It's easy to see, the following theorem holds.

Theorem 9: (Integrity of MAC-and-Encrypt with a pseudorandom permutation encryption) Let SE be a pseudorandom permutation encryption scheme, let MA be a message authentication scheme and let $\overline{SE}$ be the encryption scheme obtained from SE and MA via the Definition 5. If MA is SUF-CMA-secure, then $\overline{SE}$ is INT-CTXT secure.

Concretely, for any adversary F

$$Adv_{EJM}^{int-ctxt}(F) \leq Adv_{MA}^{suf-cma}(F) \quad (10)$$

Remark: Actually, the Theorem 9 holds for many modes of the underlying encryption. One example is the OTP mode, which leads any change of a bit in the ciphertext equivalent to the change of the corresponded bit in the plaintext. We refer the other two examples to the famous mode of CBC and CTR (R-CTR as well as C-CTR), as long as the initial vector IV or the initial value of the counter is dealt along with the plaintext by the MAC.

CONCLUSION

In this study, we discussed the other side of authentication techniques, presented a new security

notation of authentication that we call it the tag secrecy. The positive result is that, for some common modes of encryption, CBC (with a secure underlying block cipher) and OTP (stream ciphers that xor data with a pseudorandom pad) or CTR, the tag secrecy of MAC enables the MAC-and-Encrypt paradigm to implement secure channels.

REFERENCES

1. Ylonen, T., T. Kivinen, M. Saarinen, T. Rinne and S. Lehitnen, 2001. SSH Trasport Layer Protocol, http://www.ietf.org/internet-drafts/draft-ieft-secsh-transport-09.txt
2. Ylonen, T., 2005. SSH transport layer protocol. http://www.ietf.org/internet-drafts/draft-ietf-secsh-transport-24.txt
3. Krawczyk, H., 2001. The Order of Encryption and Authentication for Protecting Communications (or: How secure is SSL?). In J. Kilian, Editor, Advances in Cryptology-CRYPTO 2001, Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 2139: 310-331.
4. Bellare, M. and C. Namprempre, 2000. Authenticated Encryption: Relations Among Notions and Analysis of the Generic Composition Paradigm. In T. Okamoto, editor, Advances in Cryptology-ASIACRYPT 2000, Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1976: 531-545.
5. Canetti, R. and H. Krawczyk, 2001. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels, Advances in Cryptology- EUROCRYPT 2001 Proceedings, Lecture Notes in Computer Science, Springer-Verlag, B. Pfitzmann, Edn., Full version in: Cryptology eprint Archive (http: //eprint. iacr. org/), Report, 2001/040. 2045: 453-474.
6. Bellare, M., T. Kohno and C. Namprempre, 2004. Breaking and provably repairing the SSH authenticated encryption scheme: A Case Study of the Encode-then-MAC-and-Encrypt Paradigm, ACM Transactions on Information and System Security, ACM.
7. Goldwasser, S. and M. Bellare, 2001. Lecture Notes on Cryptography. Http://theory.lcs.mit.edu/shefi, pp: 103-104.
8. Goldwasser, S. and M. Bellare, 2001. Lecture notes on cryptography, Http://theory.lcs.mit.edu/shefi, pp: 141,149.
9. Bellare, M., A. Desai, E. Jokipii and P. Rogaway, 1997. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. In Proc. 38th IEEE Symp. on Foundations of Comp. Science. IEEE.
10. Goldwasser, S. and M. Bellare, 2001. Lecture notes on cryptography, Http://theory.lcs.mit.edu/shefi, pp: 161-163.
11. Bellare, M., R. Canetti and H. Krawczyk, 1996. Keying hash functions for message authentication. In Proceedings of Crypto 96, vol. 1109 of Lecture Notes in Computer Science. Springer-Verlag.
12. Rfc2104, HMAC. Keyed-Hashing for Message Authentication. http: //www. Faqs.Org/frcs/rfc2104. htm1.