FORWARD-SECURE AUTHENTICATED-ENCRYPTION IN MULTI-RECEIVER SETTING

Kan Yasuda, Kazumaro Aoki, Eiichiro Fujisaki and Atsushi Fujioka
NTT Information Sharing Platform Laboratories, Nippon Telegraph and Telephone Corporation
1-1 Hikarinooka Yokosuka-shi, Kanagawa-ken 239-0847 Japan

Keywords: Authenticated encryption, strong integrity, multi-receiver setting, forward security, packet-loss detection, DoS attack, message authentication code, pseudo-random bit generator, target-collision-resistant hash-function.

Abstract: In this paper we study a generic construction of forward-secure authenticated-encryption in unidirectional, multi-receiver setting. By "multi-receiver setting" we mean the situation in which a single center transmits large data to a dynamically changing group of receivers. In such scenario a direct application of bidirectional, unicast methods would lead to multiple problems. In particular, we focus on the problem of vulnerability against a type of denial-of-service (DoS) attack. We show that this problem can be effectively resolved by a mechanism we call "MAC-then-MAC" structure. As the name suggests, this structure uses two independent MACs, but we illustrate how it can be realized without losing efficiency in transmission rate, storage size and computational overhead. Despite the fact that one of the MACs uses a constant key, our construction guarantees integrity in the sense of forward security. We provide a concrete security model showing that our construction achieves confidentiality and strong integrity (replay avoidance, in-order packet delivery, etc.) both in the sense of forward security.

1 INTRODUCTION

The area of *authenticated encryption* deals with symmetric-key techniques to provide both confidentiality and integrity. It can be realized by a generic composition of a symmetric-key cipher and a message authentication code (MAC) (Bellare and Namprempre, 2000), which works well in unicast setting. On the other hand, in *multi-receiver setting*, where a single center transmits large data to a dynamically changing group of receivers, multiple problems arise in constructing authenticated encryption (Note that in such scenario the communication becomes unidirectional from the service provider to each user, after secret keys are once distributed to users.) One problem is inefficiency; the direct application of unicast authenticated-encryption would cause overhead in transmission rate and in center's computation. Another problem is that the unidirectional aspect demands the mechanism of *packet-loss detection* in passive mode, in addition to other integrity requirements such as replay avoidance and in-order packet delivery.

In this paper we focus on another problem in realizing authenticated encryption in multi-receiver setting: *forward security*. Note that in multi-receiver setting, secret keys, as many as the number of users, are involved. These keys are long-lived, so they are always under the risk of being compromised.

In general, the forward security of authenticated encryption can be realized by combining a pseudo-random number/bit generator (PRNG) and updating the secret keys periodically (Bellare and Yee, 2003). In multi-receiver setting, however, its unidirectional aspect makes it difficult to update the keys in synchrony between the center and each user. The unidirectional transmission also leaves the PRNG+MAC method (which assures the integrity well in the sense of forward security in bidirectional communication) vulnerable against a type of denial-of-service (DoS) attack.

The contribution of this paper is to provide a concrete, provably-secure construction of authenticated encryption that resist against this DoS attack. The DoS attack is overcome by a generic construction we call "MAC-then-MAC" structure. This structure, as its name suggests, uses two independent MACs and hence would introduce inefficiency in transmission rate, storage (key) size and computational overhead. Yet, it turns out that the MAC-then-MAC structure possesses a property with which we can suppress the

Yasuda K., Aoki K., Fujisaki E. and Fujioka A. (2006).
FORWARD-SECURE AUTHENTICATED-ENCRYPTION IN MULTI-RECEIVER SETTING.
In *Proceedings of the International Conference on Security and Cryptography*, pages 141-148
DOI: 10.5220/0002101501410148

increase in bandwidth and storage size. The computational inefficiency can be resolved by an introduction of keyed hash-function. We also show that our construction attains confidentiality and strong integrity in the sense of forward security, even though one of the MACs uses a constant key.

ORGANIZATION OF THIS PAPER. In Section 2 we provide the background and previous works for the topics discussed in this paper. In Section 3 we describe a naive construction based on previous techniques, which assures confidentiality and strong integrity in the sense of forward security but is vulnerable against the DoS attack. Then we provide an improved construction in Section 4 with the MAC-then-MAC structure, which resists against the DoS attack without losing efficiency. Precise definitions of our algorithms are given in Section 5. Sections 6 and 7 are devoted for the security analysis of our scheme. Our security proofs are conducted in the *concrete security* model (Bellare et al., 1997). This model is more suited to symmetric-key setting than the asymptotic, polynomial-reduction security model, for in practice a symmetric-key primitive is usually equipped with a fixed security parameter. In Section 8 we discuss practical instantiations of our construction. Section 9 concludes this paper.

2 PRELIMINARIES

RELATED WORKS. (Park et al., 2002) studies efficient approaches for authentication in multicast, combining a digital signature with other techniques. In particular, their construction is robust against packet loss and suited for authenticating real-time streamed data (Golle and Modadugu, 2001). There are also works that deal with efficient methods for key management (Wong et al., 2000). Among them, there is an area called "broadcast encryption" (Naor et al., 2001) which focuses on the confidentiality aspect and gives efficient methods for key distribution and mechanisms of traitor tracing. Practical services that fit into this setting include the distribution of copyright-protected materials. Lastly, we mention the recent work (Ray et al., 2005) which gives an RSA-based multicast scheme with an added feature of anonymity. Most of the above works either depend on asymmetric-key techniques or involve rather sophisticated ways to manage secret keys, and converting these mechanisms into the form of forward security does not seem trivial. So we do not go into the details here.

BASIC APPROACH (TWO-STAGE ENCRYPTION). In this paper we adopt a straight-forward approach for key distribution, as follows: We assume that the center distributes an independently random secret-key k_u to each user u prior to transmission of data. Then the center uses a fresh, random session-key K for each message m. m is encrypted via a symmetric-key cipher $E : \{0,1\}^{\kappa_E} \times \{0,1\}^* \to \{0,1\}^*$ as $c \leftarrow E_K(m)$, and K is encrypted via another symmetric-key cipher $\bar{E}$ as $\vec{h} \leftarrow \bar{E}_{\vec{k}}(K)$ (here the vector notations mean $h_u \leftarrow \bar{E}_{k_u}(K)$ for each u), where $\vec{k} = (k_u)_{u \in P}$ is the vector of long-lived secret keys for the set P of permitted users. Then the encrypted datum $(c, \vec{h})$ consisting of ciphertext and header is delivered to the permitted receivers. This two-stage encryption improves the efficiency (as compared to encrypting the large message m with each k_u) and at least assures confidentiality. This method is suited for a situation with a relatively small number of users and provides a practically efficient solution as long as the size of header $\vec{h}$ is kept minimal.

PRACTICAL SERVICES. In multi-receiver scenario considered in this paper, a user may be revoked or re-joined upon the decision of the sender; the center may update the list of recipients dynamically during a sequence of transmissions. Practical services that fit into this setting include the distribution of mail magazine and the multicast of pay contents: A user may wish to be unsubscribed from the mail magazine any time during the service, or the service provider may wish to default temporarily the transmission of contents to those who have failed to pay.

AUTHENTICATED ENCRYPTION ("ENCRYPT-THEN-MAC"). A mechanism of authenticated encryption can be realized either by a dedicated scheme or by a generic composition of a symmetric cipher and a MAC (Bellare and Namprempre, 2000). Applicability of dedicated schemes to multi-receiver setting heavily depends on the structure of each scheme, while the generic composition paradigm gives us scalability. So we adopt the well-known "Encrypt-then-MAC" composition (Bellare and Namprempre, 2000) as our starting point.

STRONG INTEGRITY AND COUNTER. The traditional notion of integrity deals with protecting data contents from being modified. An appropriate usage of a MAC would be adequate for this purpose. However, this does not suffice to guarantee the strong integrity required in practice, such as replay avoidance and in-order packet delivery (Kohno et al., 2003). The strong integrity is a highly desirable property in multi-receiver services.

The strong integrity of a sequence of ciphertexts can be assured in several ways. Among them is to assign a *counter* (like the "sequence number" used in SSL) to each ciphertext (Note that the counter can be sent in the clear. Also note that the size of the counter, which correspondingly establishes its range, must be large enough to meet the security objectives.

Preferably, it should be as large as the security parameter.) This counter method works effectively even in the unidirectional setting, so we adopt this method as our starting point.

PACKET-LOSS DETECTION. In this paper we require each user u's ability for packet-loss detection in a passive manner. This mechanism is quite important in multi-receiver scenario, as explained in the following: Suppose the center transmits ciphertexts $C_1, \ldots, C_n$, in this order. Then some intermediate C_i may not be delivered correctly to a permitted user u, for multiple reasons. One possible reason is that noises on the network may cause a damage to C_i, resulting u to reject the received ciphertext. Another possible situation is the case of mail magazine; the receiving server (mail transfer agent, MTA) for u may happen to be out of service on transferring C_i. Upon failure, an error code may be sent back (automatically) by the MTA to the service provider, but the service provider may not care for re-sending the old data. In case of stateful encryption, it may be even impossible for the service provider to re-send the exactly same ciphertext C_i. In these cases it is highly desirable that the user u be able to detect the packet loss for C_i, upon decrypting the following (intact) ciphertext C_{i+1}. We also note that it is too optimistic to hope that $C_1, \ldots, C_n$ reach every user in the correct order, when we think of the evolving complexities of routing architecture in the Internet.

FORWARD SECURITY VIA PRNG. The forward security of authenticated encryption can be realized by key-update via a stateful PRNG (Bellare and Yee, 2003). If the combined PRNG is forward secure, then the resulting authenticated-encryption scheme attains forward security of both confidentiality and (traditional) integrity (Bellare and Yee, 2003).

A stateful PRNG $G : \{0,1\}^{\sigma_G} \to \{0,1\}^{\sigma_G+\kappa_G}$ is a deterministic algorithm that takes as its input the current state s_i and outputs $(s_{i+1}, k_{i+1}) \leftarrow G(s_i)$, where s_{i+1} is the next state and k_{i+1} the updated key. The secret key k_i is used by the underlying symmetric-key cipher and MAC. Note that the current stage number i must be attached to the ciphertext/tag and sent in the clear, so that the receiver can update his key in synchrony and decrypt/verify with the key of correct stage.

LIFE CYCLE OF SECRET KEYS. The length of a session for using the same secret key k_i must be agreed between the sender and the receiver, so that the communicating party can update the secret key in synchrony. The frequency of key update reflects the policy of the service and may be decided in terms of either the time-span or the number of encrypted messages. The former is not suited in multi-receiver setting, because a user may be excluded from the service for a long period of time. The latter is more suited, but note that the center needs to maintain the record of the number of messages encrypted with the current key k_i for each user u.

VULNERABILITY AGAINST DOS ATTACK. The PRNG+MAC method experiences the following DoS attack in unidirectional setting: The adversary sends to u bogus ciphertexts $C'_1, C'_2, \ldots$ with stage numbers $i'^{(u)}_1, i'^{(u)}_2, \ldots \gg i^{(u)}$, where $i^{(u)}$ is the current stage number of u. Then u needs to investigate each C'_j if its contents are forged or left intact, so that in the latter case u can store C'_j and wait for missing ciphertexts between $i^{(u)}$ and $i'^{(u)}$ to arrive.

If u performs the on-the-fly integrity-check for each C'_j, then each integrity-check requires $(i'^{(u)}_j - i^{(u)})$-many invocations of G, expending u's computational resources. If u simply stores every C'_j into memory without integrity-check, then it would flood u's memory. If u saves intermediate state/key information into memory upon decrypting the first C'_1, then it would still need huge memory and undesirably increase the risk of state/key compromise.

3 NAIVE COMPOSITION: TWO-STAGE ENCRYPTION + "ENCRYPT-THEN-MAC" + COUNTER + PRNG

Figure 1 illustrates a naive construction of encryption algorithm, which is vulnerable against the DoS attack but provides a prototype for our construction. It is based on the well-known "Encrypt-then-MAC" composition (Bellare and Namprempre, 2000) applied to the two-stage encryption via symmetric-key ciphers E and $\bar{E}$ with a MAC $\bar{T} : \{0,1\}^{\kappa_{\bar{T}}} \times \{0,1\}^* \to \{0,1\}^\tau$. Each user u's counter $\mathsf{ctr}^{(u)}$ is attached to the ciphertext in order to attain the strong integrity. The long-lived secret keys $k^{(u)}_{\bar{E}}$ for $\bar{E}$ and $k^{(u)}_{\bar{T}}$ for $\bar{T}$ are updated via PRNG G (Bellare and Yee, 2003) for the purpose of forward security. G operates as $(s^{(u)}, k^{(u)}_{\bar{E}}, k^{(u)}_{\bar{T}}) \leftarrow G(s^{(u)})$ and is associated with the stage number $i^{(u)}$. In Figure 1 the variables with a superscript "(u)" are unique to each receiver u, and those without superscripts are common to all receivers.

The encryption algorithm $\mathcal{E}$ takes as its input the message $m \in \{0,1\}^*$ and outputs the ciphertext $C = (c, \vec{h})$ with $\vec{h} = (h^{(u)})_{u \in P}$. u's header $h^{(u)}$ consists of $\mathsf{ctr}^{(u)} \| i^{(u)} \| \hat{h}^{(u)} \| \overline{\mathsf{tag}}^{(u)}$, and P denotes the set of permitted users.

In the following we list some (negative) aspects of this construction:

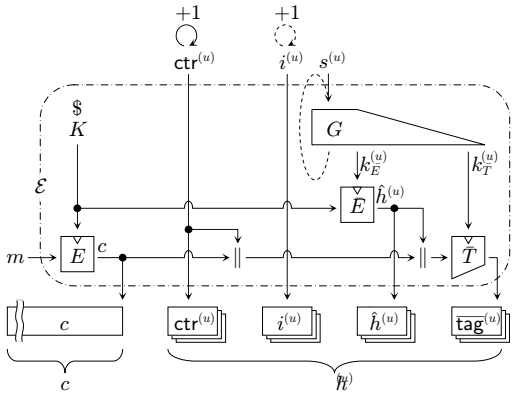

Figure 1: Naive construction of encryption algorithm.

STRONG REQUIREMENT FOR $\bar{E}$. Under chosen-plaintext attacks[1] against this scheme, the symmetric-key cipher $\bar{E}$ needs to be secure against known-plaintext attacks; the long-lived secret key $k_{\bar{E}}^{(u)}$ is used for encrypting multiple session-keys K, even though the adversary cannot choose the input K. Note that, on the other hand, the symmetric-key cipher E only needs to be one-time secure, because the session-key K is used for encrypting only a single message.

OVERHEAD IN COMPUTING TAGS. In Figure 1 the center needs to compute as many tags $\overline{\mathsf{tag}}^{(u)}$ as the number of receivers, with the large ciphertext c as a part of the input. Inputting c into the MAC $\bar{T}$ with each user's secret key $k_{\bar{T}}^{(u)}$ is necessary, because MACing instead with a common (fresh, random) session-key would lead to a trivial attack by a revoked user.

INDIVIDUAL COUNTER $\mathsf{ctr}^{(u)}$. In Figure 1 the center needs to assign an individual counter $\mathsf{ctr}^{(u)}$ to each user u. The assignment of individual counter is necessary for the packet-loss detection and for the management of key-update frequency.

Note that the usage of a counter common to all users would no longer guarantee the users' ability for packet-loss detection in a passive manner; the receiver cannot distinguish if the missing ciphertexts are lost due to (adversarial) network damage or if they are skipped due to the fact that the receiver in question has been excluded during the period. It would also disable the management of life cycles of secret keys.

INDIVIDUAL STAGE-NUMBER $i^{(u)}$. The individual stage number $i^{(u)}$ also increases the size of header $h^{(u)}$ and the center's storage. Note that if one uses a common stage number, then a long-revoked user, upon re-joining, must perform the computation of G many times in order to obtain the current secret-key.

STORAGE SIZE. Figure 2 shows the storage at the center. It can be minimized down to $(\mathsf{ctr}^{(u)}, s^{(u)})_u$, if $i^{(u)}$ is deduced from $\mathsf{ctr}^{(u)}$ and $k_{\bar{E}}^{(u)}, k_{\bar{T}}^{(u)}$ by $s^{(u)}$.

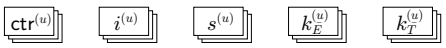

Figure 2: Storage with the naive construction.

4 PROPOSED SOLUTION

In this section we modify the construction described in Section 3, introducing the MAC-then-MAC structure. Our improved encryption algorithm $\mathcal{E}$ is described in Figure 3.

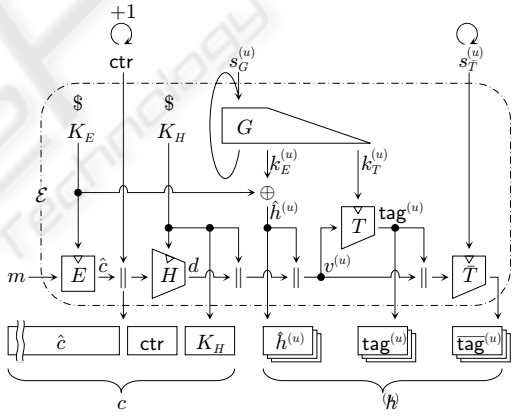

Figure 3: Improved construction of encryption algorithm.

In the following we list some of the key points in this construction:

KEY-UPDATE UPON EVERY ENCRYPTION. In Figure 3 we simply require the encryption algorithm $\mathcal{E}$ to update the secret keys $k_E^{(u)}$ and $k_T^{(u)}$ (of targeted receivers u) upon every invocation. This resolves the center's cumbersome management. This also reduces the storage at the center, for the service provider needs to save only the state $s_G^{(u)}$ and not these keys.

Updating the secret keys on each encryption may seem to increase the computational overhead at the center, but this increase is cancelled by the replacement of $\bar{E}$ with XOR ($\oplus$); note that since the keys for $\bar{E}$ are now updated each time, $\bar{E}$ need to be only one-time secure.

[1] It suffices to consider only the chosen-plaintext attacks here, because we shall consider the integrity requirements separately. The security against chosen-plaintext/ciphertext attacks would then follow immediately, provided that those returned by the encryption oracle are never queried to the decryption oracle.

INTRODUCTION OF KEYED HASH-FUNCTION H. The hash function $H : \{0,1\}^{\kappa_H} \times \{0,1\}^* \to \{0,1\}^{\delta}$ in Figure 3 reduces the center's overhead in computing tags. H is keyed with a (fresh, random) session-key K_H, which is common to all receivers.

"MAC-THEN-MAC" STRUCTURE. We use two MACs in Figure 3. The external MAC $\bar{T}$ uses a constant, long-lived secret key $s_{\bar{T}}^{(u)}$, and the internal MAC $T : \{0,1\}^{\kappa_T} \times \{0,1\}^* \to \{0,1\}^{\tau}$ uses a key $k_T^{(u)}$ that is updated via G periodically (upon every encryption). The external MAC $\bar{T}$ gets around with the DoS attack and enables us to omit the stage number $i^{(u)}$, while the internal MAC T assures forward security and enables us to use a common counter ctr. So this "MAC-then-MAC" mechanism doubles the size of tag but keeps minimal the increase in the header size. Also note that these MACs T and $\bar{T}$ now have fixed-size inputs, owing to the hash function H. In Section 6 it will become clear why the usage of constant key for the outer MAC still guarantees forward security.

STORAGE SIZE. Figure 4 shows the storage at the center. Note that the increase is just ctr, under the assumption $\left|\mathsf{ctr}^{(u)}\right| = \left|s_{\bar{T}}^{(u)}\right|$.

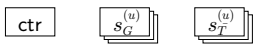

Figure 4: Storages with the improved construction.

5 ALGORITHM DESCRIPTION

Our authenticated-encryption scheme $\mathcal{AE}$, based on the improved construction in Section 4, is encapsulated into three algorithms as $\mathcal{AE} = (\mathcal{S}, \mathcal{E}, \mathcal{D})$. In this section we give precise definitions of these algorithms.

STATE GENERATION ALGORITHM. $\mathcal{S}$ is a probabilistic algorithm that takes as its input the set U of all users[2] and outputs the initial state $S = (\mathsf{ctr}, \vec{s}) \leftarrow \mathcal{S}(U)$, where ctr is the state information common to all users (i.e., the counter) and $\vec{s} = (s^{(u)})_{u \in U}$ the vector of users' secret information. We use the notation $S^{(u)} = (\mathsf{ctr}, s^{(u)})$, and the state $S^{(u)}$ is distributed to the user u prior to transmission of data.

Algorithm $\mathcal{S}(U)$
$\mathsf{ctr} \leftarrow 0$;
$s^{(u)} \xleftarrow{\$} \{0,1\}^{\kappa_G + \kappa_{\bar{T}}}$ for each $u \in U$;
$\vec{s} \leftarrow (s^{(u)})_{u \in U}$; $S \leftarrow (\mathsf{ctr}, \vec{s})$
Return S

[2]For simplicity we fix the set U of all users, but in practice a new user can be freely joined at any time during the service.

ENCRYPTION ALGORITHM. $\mathcal{E}$ is invoked by the center. $\mathcal{E}$ is a probabilistic and stateful algorithm that takes as its input the current state S_i (which contains the counter plus all users' secret information), a recipient list P and a plaintext m, and that outputs $(S_{i+1}, C) \leftarrow \mathcal{E}(S_i, P, m)$. S_{i+1} denotes the next state. C denotes the encrypted data $(c, \vec{h})$, where c is the common ciphertext and $\vec{h} = (h^{(u)})_{u \in P}$ the header vector. We use the notation $C^{(u)} = (c, h^{(u)})$. Upon receiving $C = (c, \vec{h})$, each user u parses the data and retrieves $C^{(u)}$. The user u is concerned about confidentiality and integrity of data $C^{(u)}$ only (and not the entire C).

Algorithm $\mathcal{E}(S, P, m)$
$K_E \xleftarrow{\$} \{0,1\}^{\kappa_E}$; $\hat{c} \leftarrow E_{K_E}(m)$;
$(\mathsf{ctr}, \vec{s}) \leftarrow S$; $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$; $K_H \xleftarrow{\$} \{0,1\}^{\kappa_H}$;
$d \leftarrow H_{K_H}(\hat{c} \| \mathsf{ctr}) \| K_H$
For each $u \in P$ do:
Parse $s^{(u)} = s_G^{(u)} \| s_{\bar{T}}^{(u)}$;
$(s_G^{(u)}, k_E^{(u)}, k_T^{(u)}) \leftarrow G(s_G^{(u)})$; $s^{(u)} \leftarrow s_G^{(u)} \| s_{\bar{T}}^{(u)}$;
$\hat{h}^{(u)} \leftarrow k_E^{(u)} \oplus K_E$; $v^{(u)} \leftarrow d \| K_H \| \hat{h}^{(u)}$;
$\mathsf{tag}^{(u)} \leftarrow T_{k_T^{(u)}}(v^{(u)})$;
$\overline{\mathsf{tag}}^{(u)} \leftarrow \bar{T}_{s_{\bar{T}}^{(u)}}(v^{(u)} \| \mathsf{tag}^{(u)})$;
$h^{(u)} \leftarrow \hat{h}^{(u)} \| \mathsf{tag}^{(u)} \| \overline{\mathsf{tag}}^{(u)}$
EndFor
$\vec{s} \leftarrow (s^{(u)})_{u \in P}$; $S \leftarrow (\mathsf{ctr}, \vec{s})$;
$\vec{h} \leftarrow (h^{(u)})_{u \in P}$; $C \leftarrow (c, \vec{h})$
Return (S, C)

DECRYPTION ALGORITHM. $\mathcal{D}$ is associated with each user u. $\mathcal{D}$ is a deterministic and stateful algorithm that takes as its input u's current state $S_i^{(u)}$ and received ciphertext $C^{(u)}$, and outputs $(S_{i+1}^{(u)}, m) \leftarrow \mathcal{D}(S_i^{(u)}, C^{(u)})$. $\mathcal{D}$ outputs $m = \perp$ when rejecting $C^{(u)}$ and $m = \curvearrowright$ when detecting packet loss. In these cases the state does not get updated; i.e., $S_{i+1}^{(u)} = S_i^{(u)}$. Otherwise, $\mathcal{D}$ outputs the next state $S_{i+1}^{(u)}$ and the plaintext m.

Algorithm $\mathcal{D}(S^{(u)}, C^{(u)})$
$(\mathsf{ctr}, s^{(u)}) \leftarrow S^{(u)}$; Parse $s^{(u)} = s_G^{(u)} \| s_{\bar{T}}^{(u)}$;
$(c, h^{(u)}) \leftarrow C^{(u)}$
Parse $h^{(u)} = \hat{h}^{(u)} \| \mathsf{tag}^{(u)} \| \overline{\mathsf{tag}}^{(u)}$
Parse $c = \hat{c} \| \mathsf{ctr}' \| K_H$; $d \leftarrow H_{K_H}(\hat{c} \| \mathsf{ctr}')$;
$v^{(u)} \leftarrow d \| K_H \| \hat{h}^{(u)}$;
$\overline{\mathsf{tag}}'^{(u)} \leftarrow \bar{T}_{s_{\bar{T}}^{(u)}}(v^{(u)} \| \mathsf{tag}^{(u)})$;
$(s_G'^{(u)}, k_E^{(u)}, k_T^{(u)}) \leftarrow G(s_G^{(u)})$;
$\mathsf{tag}'^{(u)} \leftarrow T_{k_T^{(u)}}(v^{(u)})$
If $\mathsf{ctr}' \not> \mathsf{ctr}$ or $\overline{\mathsf{tag}}'^{(u)} \neq \overline{\mathsf{tag}}^{(u)}$
then return $(S^{(u)}, \perp)$

Else
If $\mathsf{tag}'^{(u)} \neq \mathsf{tag}^{(u)}$ then return $(S^{(u)}, \curvearrowright)$
Else
$K_E \leftarrow k_E^{(u)} \oplus \hat{h}^{(u)}$; $m \leftarrow E_{K_E}^{-1}(\hat{c})$;
$s'^{(u)} \leftarrow s_G'^{(u)} \| s_{\bar{T}}^{(u)}$; $S'^{(u)} \leftarrow (\mathsf{ctr}', s'^{(u)})$
Return $(S'^{(u)}, m)$
EndIf
EndIf

CONSISTENCY REQUIREMENTS. Initialize the state as $S_0 \leftarrow \mathcal{S}$. For a sequence $(P_0, m_0), \ldots, (P_n, m_n)$ set $(S_{i+1}, C_i) \leftarrow \mathcal{E}(S_i, P_i, m_i)$ for $i = 0, \ldots, n$. For $u \in U$ let $i_0^{(u)}, i_1^{(u)}, \ldots$ denote the indices of P to which u belongs, in increasing order. Set $S_0' \leftarrow S_0$ and $(S_{j+1}'^{(u)}, m_j'^{(u)}) \leftarrow \mathcal{D}(S_j'^{(u)}, C_{i_j^{(u)}}^{(u)})$ for $j = 0, 1, \ldots$. We require the soundness conditions as follows: For all u and j such that $u \in P_{i_j^{(u)}}$ the condition $m_j'^{(u)} = m_{i_j^{(u)}}$ must hold. Also, for all u and $j < j'$ such that $u \in P_{i_j^{(u)}}, P_{i_{j'}^{(u)}}$ the conditions $(S_{j'}'^{(u)}, \perp) = \mathcal{D}(S_{j'}'^{(u)}, C_{i_j^{(u)}}^{(u)})$ and $(S_j'^{(u)}, \curvearrowright) = \mathcal{D}(S_j'^{(u)}, C_{i_{j'}^{(u)}}^{(u)})$ must hold.

6 SECURITY DEFINITIONS

In the following we describe the adversarial models we adopt in our security analysis.

CONFIDENTIALITY. (Bellare and Yee, 2003) discussed the forward security of symmetric-key encryption and introduced an adversarial model based on *find-then-guess* indistinguishability. However, this adversarial model is not amenable to our stateful encryption, for in our case each state is used only once to encrypt a single message and never used again to encrypt another one. So instead we adopt the security notion *real-or-random* indistinguishability (Bellare et al., 1997) and modify it into the form of forward security. This is essentially modeled on the notion of forward-secure PRNGs (Bellare and Yee, 2003).

We modify the adversarial model into a form that accords to multi-receiver settings. Namely, for a challenge bit $b \in \{0,1\}$ and an adversary A attacking an authenticated-encryption scheme $\mathcal{AE} = (\mathcal{S}, \mathcal{E}, \mathcal{D})$, we consider the following experiment:[3]

Experiment $\mathbf{Exp}_{\mathcal{AE}}^{\text{fsind-cpa-}b}(A)$
$S \leftarrow \mathcal{S}$
Run $A^{\mathcal{O}_S(\cdot,\cdot)}$
Reply to $\mathcal{O}_S(P, m_1)$ as follows:
If $A \in P$ then
$(S, C) \leftarrow \mathcal{E}(S, P, m_1)$; $A \Leftarrow C$ EndIf
If $A \notin P$ then $m_0 \xleftarrow{\$} \{0,1\}^{|m_1|}$;
$(S, C) \leftarrow \mathcal{E}(S, P, m_b)$; $A \Leftarrow C$ EndIf
Until A outputs intrude
$A \Leftarrow S^{(u)}$ for each $u \in \bar{P}$; $\tilde{b} \leftarrow A$
Return $\tilde{b}$

In the above experiment, queries with $A \in P$ are always answered with real, and those with $A \notin P$ are answered with real or random, depending on the value b. At the end of queries to $\mathcal{O}$, the adversary A outputs intrude and is fed with the current state $S^{(u)}$ of users $u \in \bar{P}$. We assume that A's access set $\bar{P}$, such that $P \subset \bar{P}$ for all queries (P, m_1), is known a priori.

We then measure the adversary A's advantage via

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{fsind-cpa}}(A) = \Pr\left[\mathbf{Exp}_{\mathcal{AE}}^{\text{fsind-cpa-1}}(A) = 1\right] - \Pr\left[\mathbf{Exp}_{\mathcal{AE}}^{\text{fsind-cpa-0}}(A) = 1\right].$$

Also we define the advantage function as

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{fsind-cpa}}(t, q_{\text{in}}, q_{\text{out}}, \mu, \pi) = \max_A \mathbf{Adv}_{\mathcal{AE}}^{\text{fsind-cpa}}(A),$$

where the maximum is taken over all adversaries A with time complexity at most t (including its code size), each making at most q_{in} queries to the $\mathcal{O}$ oracle with $A \in P$ and q_{out} with $A \notin P$, with each m_1 being at most μ bits and the cardinality of $\bar{P}$ being at most π.

INTEGRITY. We basically adopt the strongest "Type5" integrity notion (Kohno et al., 2003), with an enhancement for packet-loss detection. We shall modify the adversarial model into the form of forward security (Bellare and Yee, 2003) and multi-receiver setting:

Experiment $\mathbf{Exp}_{\mathcal{AE}}^{\text{fsint-ctxt}}(A)$
$S \leftarrow \mathcal{S}$; $S' \leftarrow S$; $S'' \leftarrow S$
$(i^{(u)})_{u \in U} \leftarrow \vec{0}$; $(i'^{(u)})_{u \in U} \leftarrow \vec{0}$; $(i''^{(u)})_{u \in U} \leftarrow \vec{0}$
Run $A^{\mathcal{O}_S(\cdot,\cdot), \mathcal{O}_{S'}^{-1}(\cdot,\cdot)}(\mathsf{before})$
Reply to $\mathcal{O}_S(P, m)$ as follows:
$(S, C) \leftarrow \mathcal{E}(S, P, m)$;
$i^{(u)} \leftarrow i^{(u)} + 1$ for each $u \in P$;
$C_{i^{(u)}}^{(u)} \leftarrow C^{(u)}$ for each $u \in P$; $A \Leftarrow C$
Reply to $\mathcal{O}_{S'}^{-1}(u, \tilde{C}^{(u)})$ as follows: # $u \neq A$
$(S'^{(u)}, m) \leftarrow \mathcal{D}(S'^{(u)}, \tilde{C}^{(u)})$
If $m \neq \perp, \curvearrowright$ and $(i'^{(u)} + 1 > i^{(u)}$ or
$\tilde{C}^{(u)} \neq C_{i'^{(u)}+1}^{(u)})$ then return 1 EndIf
If $m = \curvearrowright$ and $(i'^{(u)} + 2 > i^{(u)}$ or
$\tilde{C}^{(u)} \notin \{C_{i'^{(u)}+2}^{(u)}, \ldots, C_{i^{(u)}}^{(u)}\})$ then
return 1 EndIf
If $m \neq \perp, \curvearrowright$ then $i'^{(u)} \leftarrow i'^{(u)} + 1$; $A \Leftarrow 1$
Else $A \Leftarrow 0$ EndIf
Until A outputs intrude

[3] Again, it suffices to consider only the chosen-plaintext attacks here.

$A \Leftarrow S^{(u)}$ for each $u \in \bar{P}$
Run $A^{\mathcal{O}_{S''}^{-1}(\cdot,\cdot)}(\mathsf{after})$
Reply to $\mathcal{O}_{S''}^{-1}(u, \tilde{C}^{(u)})$ as follows: # $u \neq A$
$(S''^{(u)}, m) \leftarrow \mathcal{D}(S''^{(u)}, \tilde{C}^{(u)})$
If $i''^{(u)} + 1 \leq i^{(u)}$ and $m \neq \perp, \curvearrowright$ and
$\tilde{C}^{(u)} \neq C^{(u)}_{i''^{(u)}+1}$ then return 1 EndIf
If $m \neq \perp, \curvearrowright$ then $i''^{(u)} \leftarrow i''^{(u)} + 1$; $A \Leftarrow 1$
Else $A \Leftarrow 0$ EndIf
Until A halts
Return 0

In the above experiment we assume that A's access set $\bar{P}$ (such that $P \subset \bar{P}$ and $u \in \bar{P}$ for all queries (P, m) and $(u, \tilde{C}^{(u)})$) is known a priori. A's goal in the before stage is to forge a (new) ciphertext that is either accepted or considered as a valid, future ciphertext by the decryption oracle. At the end of before stage A outputs intrude (thereby giving up forging in the before stage) and is fed with the current state $S^{(u)}$ of users $u \in \bar{P}$. Then the state of each u's decryption oracle is rewound to its initial value, and the game resumes in after stage. Note that at this point A can freely forge future ciphertexts at stages after the intrusion, so now A's goal is to forge a new ciphertext that is accepted by the decryption oracle at a stage before the intrusion.

We measure the adversary A's advantage via

$$\mathbf{Adv}^{\text{fsint-ctxt}}_{\mathcal{AE}}(A) = \Pr\left[\mathbf{Exp}^{\text{fsint-ctxt}}_{\mathcal{AE}}(A) = 1\right].$$

Also we define the advantage function as

$$\mathbf{Adv}^{\text{fsint-ctxt}}_{\mathcal{AE}}(t, q_{\text{enc}}, q_{\text{before}}, q_{\text{after}}, \mu, \pi) = \max_A \mathbf{Adv}^{\text{fsint-ctxt}}_{\mathcal{AE}}(A),$$

where the maximum is taken over all adversaries A with time complexity at most t (including its code size), each making at most q_{enc} queries to the $\mathcal{O}$ oracle, q_{before} queries to the $\mathcal{O}^{-1}$ oracle in before stage for each u and q_{after} queries to the $\mathcal{O}^{-1}$ oracle in after stage for each u, with each m and $\tilde{c}$ (the ciphertext part of $\tilde{C}^{(u)}$) being at most μ bits and the cardinality of $\bar{P}$ being at most π.

SECURITY OF EACH PRIMITIVE. We quickly review the security definitions for the cryptographic primitives used in our construction. For a symmetric-key cipher $E : \{0,1\}^{\kappa_E} \times \{0,1\}^* \to \{0,1\}^*$ define

Experiment $\mathbf{Exp}^{\text{ind-cpa-}b}_E(A)$
$K \xleftarrow{\$} \{0,1\}^{\kappa_E}$
Run $A^{\mathcal{O}_K(\cdot)}$
Reply to $\mathcal{O}_K(m_1)$ as follows:
$m_0 \xleftarrow{\$} \{0,1\}^{|m_1|}$; $c \leftarrow E_K(m_b)$; $A \Leftarrow c$
Until A returns a bit $\tilde{b}$
Return $\tilde{b}$

$$\mathbf{Adv}^{\text{ind-cpa}}_E(A) = \Pr\left[\mathbf{Exp}^{\text{ind-cpa-1}}_E(A) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-cpa-0}}_G(A) = 1\right],$$
$$\mathbf{Adv}^{\text{ind-cpa}}_E(t, q, \mu) = \max_A \mathbf{Adv}^{\text{ind-cpa}}_E(A),$$

where the maximum is taken over all adversaries A with time complexity at most t, each making at most q queries to the $\mathcal{O}$ oracle and each m being at most μ bits.

For a PRNG $G : \{0,1\}^{\sigma_G} \to \{0,1\}^{\sigma_G + \kappa_G}$ define

Experiment $\mathbf{Exp}^{\text{prg-}b}_G(A)$
$x_0 \xleftarrow{\$} \{0,1\}^{\sigma_G + \kappa_G}$; $s \xleftarrow{\$} \{0,1\}^{\sigma_G}$;
$x_1 \leftarrow G(s)$; $A \Leftarrow x_b$; $\tilde{b} \leftarrow A$
Return $\tilde{b}$

$$\mathbf{Adv}^{\text{prg}}_G(A) = \Pr\left[\mathbf{Exp}^{\text{prg-1}}_G(A) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{prg-0}}_G(A) = 1\right],$$
$$\mathbf{Adv}^{\text{prg}}_G(t) = \max_A \mathbf{Adv}^{\text{prg}}_G(A),$$

where the maximum is taken over all adversaries A with time complexity at most t.

For a keyed hash-function $H : \{0,1\}^{\kappa_H} \times \{0,1\}^* \to \{0,1\}^{\delta}$, define

Experiment $\mathbf{Exp}^{\text{tcr}}_H(A)$
$m \leftarrow A$; $K \xleftarrow{\$} \{0,1\}^{\kappa_H}$; $d \leftarrow H_K(m)$;
$A \Leftarrow (K, d)$; $m' \leftarrow A$; $d' \leftarrow H_K(m')$
If $m' \neq m$ and $d' = d$ then return 1
Else return 0 EndIf

$$\mathbf{Adv}^{\text{tcr}}_H(A) = \Pr\left[\mathbf{Exp}^{\text{tcr}}_H(A) = 1\right],$$
$$\mathbf{Adv}^{\text{tcr}}_H(t, \mu) = \max_A \mathbf{Adv}^{\text{tcr}}_H(A),$$

where the maximum is taken over all adversaries A with time complexity at most t and with m, m' being at most μ bits.

For a MAC $T : \{0,1\}^{\kappa_T} \times \{0,1\}^{\mu_T} \to \{0,1\}^{\tau}$ with fixed-size input, define

Experiment $\mathbf{Exp}^{\text{suf-cma}}_T(A)$
$k \xleftarrow{\$} \{0,1\}^{\kappa_T}$; $L \leftarrow \varnothing$
Run $A^{\mathcal{O}_k(\cdot), \mathcal{O}_k^{-1}(\cdot,\cdot)}$
Reply to $\mathcal{O}_k(m)$ as follows:
$\mathsf{tag} \leftarrow T_k(m)$; $L \leftarrow L \cup (m, \mathsf{tag})$; $A \Leftarrow \mathsf{tag}$
Reply to $\mathcal{O}_k^{-1}(m, \mathsf{tag})$ as follows:
If $(m, \mathsf{tag}) \notin L$ and $\mathsf{tag} = T_k(m)$ then
return 1 Else $A \Leftarrow 0$ EndIf
Until A halts
Return 0

$$\mathbf{Adv}^{\text{suf-cma}}_T(A) = \Pr\left[\mathbf{Exp}^{\text{suf-cma}}_T(A) = 1\right],$$
$$\mathbf{Adv}^{\text{suf-cma}}_T(t, q_{\text{enc}}, q_{\text{dec}}) = \max_A \mathbf{Adv}^{\text{suf-cma}}_T(A),$$

where the maximum is taken over all adversaries A with time complexity at most t, each making at most q_{enc} queries to the $\mathcal{O}$ oracle and q_{dec} queries to the $\mathcal{O}^{-1}$ oracle.

7 SECURITY OF PROPOSED SCHEME

For confidentiality we have

$$\begin{aligned}\mathbf{Adv}^{\text{fsind-cpa}}_{\mathcal{AE}}(t, q_{\text{in}}, q_{\text{out}}, \mu, \pi) \leq \\ 8\pi \cdot (q_{\text{in}} + q_{\text{out}}) \cdot \mathbf{Adv}^{\text{prg}}_{G}(t) \\ + q_{\text{out}} \cdot \mathbf{Adv}^{\text{ind-cpa}}_{E}(t + q_{\text{in}} \cdot O(E(1^{\mu})), 1, \mu),\end{aligned}$$

and for integrity

$$\begin{aligned}\mathbf{Adv}^{\text{fsint-ctxt}}_{\mathcal{AE}}(t, q_{\text{enc}}, q_{\text{before}}, q_{\text{after}}, \mu, \pi) \leq \\ \pi \cdot \mathbf{Adv}^{\text{prg}}_{G}(t + q_{\text{enc}} \cdot O(G)) + q_{\text{enc}} \cdot \mathbf{Adv}^{\text{tcr}}_{H}(t, \mu) \\ + \pi q_{\text{enc}} \cdot \mathbf{Adv}^{\text{suf-cma}}_{T}(t + q_{\text{enc}} \cdot O(G), 1, q_{\text{after}}) \\ + \pi \cdot \mathbf{Adv}^{\text{suf-cma}}_{\overline{T}}(t, q_{\text{enc}}, q_{\text{before}}).\end{aligned}$$

The proof can be done in a standard hybrid argument. Due to space limitation, we omit the detailed proofs here. Note that our security reduction is in the concrete security model, with the adversary's resources being quantified.

8 PRACTICAL INSTANTIATION

PRODUCING K_E AND K_H. The random variables K_E and K_H can be generated by using a (forward secure) PRNG, provided that the secrecy of the seed/state is kept exclusively by the service provider. The security analysis can be conducted similarly when these two variables are replaced with pseudo-random bits.

INSTANTIATION OF KEYED HASH-FUNCTION H. In the recent work (Halevi and Krawcyzk, 2006) a new mode of operation for (keyless) hash function is proposed, which yields a target-collision-resistant hash-function with constant-sized keys for arbitrary long messages. So H can be instantiated with a keyless hash-function like SHA256 running in this mode.

9 CONCLUSION

In this paper we have explored the problem of DoS attack in forward-secure authentication in unidirectional, multi-receiver scenario. We have provided the "MAC-then-MAC" structure which resists against the DoS attack and can be combined with the straightforward method of key distribution. It remains an interesting work to investigate other constructions of forward-secure authenticated-encryption that can be combined with more efficient schemes.

REFERENCES

Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. (1997). A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 394–403. IEEE Computer Soceity Press.

Bellare, M. and Namprempre, C. (2000). Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Okamoto, T., editor, *Advances in Cryptology – ASIACRYPT 2000 Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer-Verlag.

Bellare, M. and Yee, B. S. (2003). Forward-security in private-key cryptography. In Joye, M., editor, *Topics in Cryptology – CT-RSA 2003 Proceedings*, volume 2612 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag.

Golle, P. and Modadugu, N. (2001). Authenticating streamed data in the presence of random packet loss. In *Network and Distributed System Security Symposium 2001 Proceedings*, pages 13–22. ISOC.

Halevi, S. and Krawcyzk, H. (2006). Strengthening digital signatures via randomized hashing. To appear in *Advances in Cryptology – CRYPTO 2006 Proceedings*.

Kohno, T., Palacio, A., and Black, J. (2003). Building secure cryptographic transforms, or how to encrypt and MAC. Cryptology ePrint Archive, Report 2003/177. `http://eprint.iacr.org/`.

Naor, D., Naor, M., and Lotspiech, J. (2001). Revocation and tracing schemes for stateless receivers. In Kilian, J., editor, *Advances in Cryptology – CRYPTO 2001 Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer-Verlag.

Park, J. M., Chong, E. K. P., and Siegel, H. J. (2002). Efficient multicast packet authentication using signature amortization. In *Proccedings of the 2002 IEEE Symposium on Security and Privacy*, pages 227–240. IEEE Computer Society.

Ray, I., Kim, E., McConnell, R., and Massey, D. (2005). Reliably, securely and efficiently distributing electronic content using multicasting. In Bauknecht, K., Pröll, B., and Werthner, H., editors, *EC-Web 2005 Proceedings*, volume 3590 of *Lecture Notes in Computer Science*, pages 327–336. Springer-Verlag.

Wong, C. K., Gouda, M., and Lam, S. S. (2000). Secure group communications using key graphs. *IEEE/ACM Transactions on networking*, 8(1):16–30.