Lower Bounds for Multicast Message Authentication

Dan Boneh
Stanford University
dabo@cs.stanford.edu

Glenn Durfee*
Stanford University
gdurf@cs.stanford.edu

Matt Franklin
University of Californa at Davis
franklin@cs.ucdavis.edu

Abstract

Message integrity from one sender to one receiver is typically achieved by having the two parties share a secret key to compute a Message Authentication Code (MAC). We consider the "multicast MAC", which is a natural generalization to multiple receivers. We prove that one cannot build a short and efficient collusion resistant multicast MAC without a new advance in digital signature design.

1 Introduction

We study the problem of message integrity in the context of a single source multicast. Consider a TV station, such as the Disney channel. The TV station is broadcasting to n receivers. Each receiver would like to ensure that the broadcasts are indeed coming from the Disney channel rather than from a malicious third party (who might be transmitting offensive material).

One natural approach would be to employ digital signatures. Suppose the transmitter has a secret signing key and each of the receivers has the corresponding public key. To provide message integrity the transmitter signs every message she broadcasts. No coalition of receivers can forge a message/signature pair that will fool another receiver. Although signatures provide multicast message integrity they are fundamentally an overkill solution for this problem. First, signatures are somewhat expensive to compute. Second, digital signatures provide non-repudiation: Any receiver can use the signature to prove to a third party that the message came from the transmitter. However, non-repudiation is unnecessary for message integrity.

Message integrity between two parties is usually done by sharing a secret key k between the sender and receiver. When sending a message M the sender computes a keyed hash function $\text{MAC} = H_k(M)$ and transmits the MAC along with the message. MACs are much faster than digital signatures, and do not provide non-repudiation. We seek a generalization of MACs for the multicast setting. This would be a distribution of keys to sender and receivers, and a method for tagging messages by the sender that would be convincing to every receiver. We call this primitive a "multicast MAC" (MMAC).

One simple approach for a MMAC might be to share a global secret key k between the transmitter and all n receivers. The transmitter appends $H_k(M)$ to every transmitted message M. Each receiver can then verify the MAC sent by the transmitter. This is insecure since any receiver can forge messages that will fool any other receiver.

Another simple approach is secure but inefficient. The transmitter shares a distinct secret key k_i with each of the n receiver $u_1, \ldots, u_n$. When sending a message M the transmitter computes

*Supported by a Microsoft Graduate Research Fellowship.

MMAC $= H_{k_1}(M) \| \ldots \| H_{k_n}(M)$ and transmits (M, MMAC). Each receiver u_i verifies the MMAC by using the entry that corresponds to the key k_i. This construction is secure, in the sense that no coalition of users can create a message/MMAC pair that will fool a user outside the coalition (since they do not have the outsider's MAC key). Unfortunately, the length of the MMAC is linear in the number of receivers. Hence, this construction is not very practical, even though it avoids non-repudiation.

Since none of the above solutions is perfect, it is tempting to try to build a MMAC that is as short as a signature (i.e., length independent of the number of receivers), but much more efficient. We give lower bounds that suggest that this might be a difficult task. Our main results show that if one could build practical (i.e. short) MMACs, then they could be converted into new efficient digital signature schemes. Consequently, it is unlikely that practical MMACs could be constructed without an unexpected advance in digital signature design.

We can relax our security requirement by saying that a MMAC is κ-secure if no coalition of size less than κ can fool another receiver. In Section 5 we generalize our lower bound and show that if one could build a κ-secure MMAC whose length is less than $\log_2 \sum_{i=0}^{\kappa} \binom{n}{i}$ then it could be converted into an efficient signature scheme. For small values of κ this lower bound is approximated by $O(\kappa \log n)$. This lower bound matches an upper bound construction based on pseudorandom functions due to Canetti et al. [1]. Hence our results show that for small values of κ the Canetti et al. construction is optimal.

Our results demonstrate the importance of recent constructions for practical multicast authentication [5, 17, 10, 11, 7, 12]. Some of these constructions achieve great efficiency (well beyond what is implied by our bounds) by making use of additional assumptions, such as weak time synchronization between sender and receivers [10, 11]. We emphasize that our lower bounds for MMACs suggest difficulty only for constructions that use the standard model for MACs, as described in the next section.

A fundamental result of theoretical cryptography is that a digital signature scheme can be derived from any one way function [9, 13, 2]. Since the existence of a multicast MAC implies the existence of a one way function, that would seem to imply a reduction of the form that we claim. However, this construction is far too inefficient to be considered for any practical purposes. In contrast, our results are achieved through direct reductions from multicast MACs to public key signature schemes. Our reductions are efficient, in the sense that the derived signature schemes have almost the same level of security as the underlying MMAC schemes.

1.1 Related work

Previous work on multicast authentication followed two tracks: (1) the computational model, based on pseudorandom functions and hash functions, and (2) the information theoretic model, providing unconditional security. Constructions in the information theoretic model provide very strong security guarantees. This strong security comes at a price: The secret key can only be used for a small number of messages. MMACs built in the computational model are not as strong, since their security depends on a complexity assumption. However, computational MMACs can be used to authenticate many messages using relatively short keys. All of the results in this paper are set in the computational model.

In the computational model, Canetti et al. [1] construct a κ-secure MMAC by concatenating many pseudorandom functions whose output is a single bit. This construction does not provide non-repudiation. As mentioned above, our results show that this clever construction is optimal. We note that the security model in [1] is slightly different from our security model. They require that a coalition should not be able to create a forgery that can fool a specific receiver. In some cases a coalition might be content if a broadcast of a forged message fools *any* receiver. Hence, in our model, a forgery is considered successful if it fools *any* receiver outside the coalition. Adapting the construction of Canetti et al. to this stronger security model adds a factor of $\ln n$ to the length of their MMAC. The result is a MMAC of length $4e(\kappa+1)\ln n \ln 1/\epsilon$ where n is the number of receivers, ϵ is the failure probability, and $e = 2.718$. For small values of κ, and a fixed ϵ, our lower bound of $O(\kappa \log n)$ asymptotically matches their upper bound.

In the information theoretic model, Multicast MACs were introduced by Desmedt, Frankel, and Yung [3] (see also Simmons [16] for the somewhat related notion of authentication codes with arbitration). They gave two constructions for κ-secure MMACs. Kurosawa and Obana [8] derived elegant lower bounds on the probability of success in impersonation and substitution attacks. They showed that the DFY construction is optimal. Safavi-Naini and Wang [14, 15] show how to construct information theoretic MMACs using cover free set systems. Their constructions are similar to the ones given in [1]. Cover free set systems were also used by Fujii, et al. [4].

We briefly review the use of signatures as an alternative to MMACs for multicast authentication. There are two difficulties in using signatures for multicast MACs: (1) in streaming audio and video applications one cannot afford to buffer the entire message prior to signing it, and (2) multicast transmissions suffer from packet loss (multicast does not provide packet loss recovery), so one needs signature schemes for an unreliable transmission channel. Problem (1) is often solved by combining standard signatures with fast one time signatures [5, 12]. Problem (2) is solved by introducing various types of redundancy during signature generation [17, 12, 10, 7].

We note that the constructions in [10, 11] provide short multicast message authentication without non-repudiation. The authentication tags in these constructions are shorter than our lower bounds predict since they rely on some weak timing synchronization between sender and receivers. Our lower bounds suggest that one must resort to such assumptions to obtain practical multicast authentication without non-repudiation.

2 Definitions

We begin by giving precise definitions for MMACs secure against existential and selective forgeries. To reduce the number of definitions in the section we only consider the strongest adversaries, namely adversaries capable of adaptive chosen message attacks. For completeness, we briefly recall definitions of security for signatures schemes.

2.1 Multicast MACs

A Multicast MAC, or MMAC, is specified by three randomized algorithms ($\mathsf{key\text{-}gen}$, $\mathsf{mac\text{-}gen}$, $\mathsf{mac\text{-}ver}$).

$\mathsf{key\text{-}gen}$: takes a security parameter s and a number of receivers n and returns keys $\mathbf{sk}, \mathbf{rk}_1, \ldots, \mathbf{rk}_n \in \{0,1\}^*$. We call $\mathbf{sk}$ the *sender key* and $\mathbf{rk}_i$ the *ith receiver key.*

$\mathsf{mac\text{-}gen}$: takes as input a message $M \in \{0,1\}^*$ and a key $K \in \{0,1\}^*$ and returns a tag $T = \mathsf{mac\text{-}gen}(M, K) \in \{0,1\}^\tau$ for some fixed tag length τ bits.

$\mathsf{mac\text{-}ver}$: takes as input a message $M \in \{0,1\}^*$, a tag $T \in \{0,1\}^\tau$, and a key $K \in \{0,1\}^*$, and returns a bit: $\mathsf{mac\text{-}ver}(M, T, K) \in \{\text{'yes'}, \text{'no'}\}$.

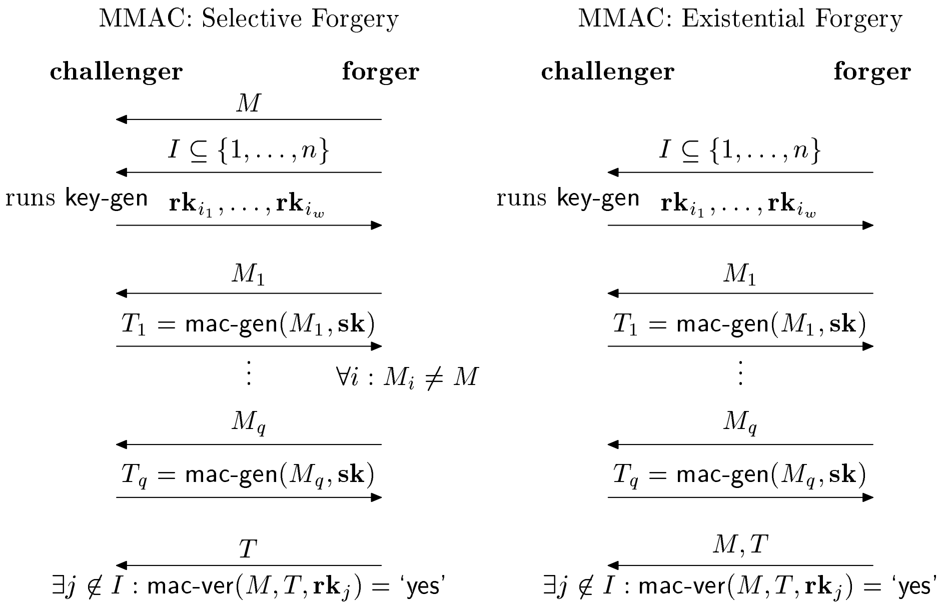

Figure 1: The games used to define two security notions for a MMAC.

These algorithms are subject to the constraint that for all ($\mathbf{sk}$, $\mathbf{rk}_1$, ..., $\mathbf{rk}_n$) produced by $\mathsf{key\text{-}gen}(s, n)$ we have that

$$\forall M \in \{0,1\}^*,\ \forall i \in \{1, \ldots, n\} : \quad \mathsf{mac\text{-}ver}(M, \mathsf{mac\text{-}gen}(M, \mathbf{sk}), \mathbf{rk}_i) = \text{'yes'}$$

In other words, tags created by mac-gen using the correct sender key verify correctly for all receivers. Each of these algorithms must run in time polynomial in n, s, and the size of the message input.

MMAC security against selective forgery

A MMAC (key-gen, mac-gen, mac-ver) is said to be $(t,\ \epsilon,\ q)$*-secure against selective forgery under an adaptive chosen message attack* if every t-time probabilistic algorithm A wins the game below with probability at most ϵ. We model the game as a communication between a challenger and the forging algorithm A. See Figure 1. We assume that the system parameters n and s are fixed ahead of time.

Step 1: The forging algorithm A starts the game by sending the challenger a target message $M \in \{0,1\}^*$. The forger's goal is to forge a MMAC for this message M. The forger also sends a subset $I \subseteq \{1, \ldots, n\}$. The subset I should be viewed as the set of receivers colluding to fool some other receiver.

Step 2: The challenger runs algorithm $\mathsf{key\text{-}gen}(s, n)$ and obtains the MMAC keys ($\mathbf{sk}$, $\mathbf{rk}_1$, ..., $\mathbf{rk}_n$). The challenger sends the subset $\{\mathbf{rk}_i\}_{i \in I}$ to A.

Step 3: Algorithm A then mounts a chosen message attack by sending queries $M_1, \ldots, M_q$ to the challenger, where $M_i \neq M$ for all $i = 1, \ldots, q$. The challenger responds with $T_i = \mathsf{mac\text{-}gen}(M_i, \mathbf{sk})$ for $i = 1, \ldots, q$. Note that these queries may be issued adaptively. That is, the adversary A might wait for a response T_i before issuing request M_{i+1}.

Step 4: Finally, A outputs a candidate MMAC, T, for the target message M.

We say that A wins this game if T verifies as a valid tag for M for some receiver j outside of I. More precisely, we say that A wins the game if

$$\exists j \notin I \qquad \text{s.t.} \qquad \mathsf{mac\text{-}ver}(M, T, \mathbf{rk}_j) = \text{`}\mathsf{yes}\text{'}.$$

The probability that A wins this game is taken over the random coin flips of the algorithms key-gen, mac-gen, mac-ver, and the random coin flips of A.

The definition above assumes the adversary commits to the set of corrupt users I at the beginning of the game. One can also consider a stronger definition where the adversary is dynamic: the adversary adaptively chooses which users to corrupt during the game. Since our lower bounds already apply when the adversary is restricted to the static settings, the same lower bounds apply in the dynamic settings. Therefore, throughout the paper we only consider static adversaries.

MMAC security against existential forgery

A MMAC (key-gen, mac-gen, mac-ver) is said to be *(t, ϵ, q)-secure against existential forgery under an adaptive chosen message attack* if every t-time probabilistic algorithm A wins the following modified game with probability less than ϵ. The game is identical to the above, except that A does not commit to the message M in Step 1. Instead, the target message M is output by A in the last step (Step 4), at the same time as the candidate tag T. Note that we must have $M \neq M_i$ for all i. See Figure 1.

2.2 Signature Schemes

Our goal is to establish a relation between MMACs and digital signatures. We therefore briefly review two notions of security for digital signatures: security against selective forgery, and security against existential forgery [6]. We review both notions under a chosen message attack.

A signature scheme is specified by three probabilistic algorithms (skey-gen, sig-gen, sig-ver).

skey-gen: takes a security parameter s and returns keys $K_{\text{sec}}, K_{\text{pub}} \in \{0,1\}^*$. We call K_{sec} the *secret key* and K_{pub} the *public key*.

sig-gen: takes as input a message $M \in \{0,1\}^*$ and a key $K \in \{0,1\}^*$ and returns a signature $S = \mathsf{sig\text{-}gen}(M, K) \in \{0,1\}^*$.

sig-ver: takes as input a message $M \in \{0,1\}^*$, a candidate signature $S \in \{0,1\}^*$, and a key $K \in \{0,1\}^*$, and returns a bit: $\mathsf{sig\text{-}ver}(M, S, K) \in \{\text{`}\mathsf{yes}\text{'}, \text{`}\mathsf{no}\text{'}\}$.

These algorithms are subject to the constraint that for all pairs (K_{sec}, K_{pub}) produced by skey-gen(s), we have that

$$\forall M \in \{0,1\}^* : \quad \mathsf{sig\text{-}ver}(M, \mathsf{sig\text{-}gen}(M, K_{\text{sec}}), K_{\text{pub}}) = \text{`}\mathsf{yes}\text{'}$$

Each of these algorithms must run in time polynomial in n, s, and the size of the input.

Signature security against selective and existential forgery

A signature scheme (skey-gen, sig-gen, sig-ver) is said to be *(t, ϵ, q)-secure against selective forgery under an adaptive chosen message attack* if every t-time probabilistic algorithm B wins the game below with probability at most ϵ. See Figure 2. We assume the security parameter s has already been fixed.

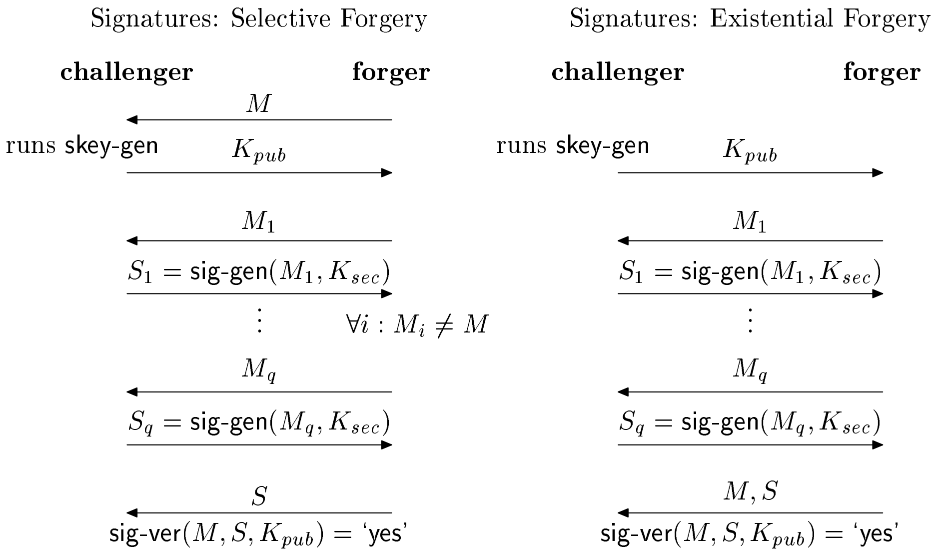

Figure 2: Signature Scheme Security.

Step 1: The forging algorithm B outputs a target message $M \in \{0,1\}^*$.

Step 2: The challenger runs algorithm skey-gen(s) and obtains the keys $(K_{\text{sec}}, K_{\text{pub}})$. The challenger sends K_{pub} to B.

Step 3: Algorithm B then mounts a chosen message attack by querying the challenger with messages $M_1, \ldots, M_q \in \{0,1\}^*$, where $M_i \neq M$ for all $i = 1, \ldots, q$. The challenger responds with $S_i =$ sig-gen(M_i, K_{sec}). Note that these queries may be issued adaptively.

Step 4: Finally, B outputs a candidate signature S for the target message M.

We say that B wins this game if S verifies as a valid signature on M. More precisely, we say that B wins this game if sig-ver$(M, S, K_{\text{pub}}) =$ 'yes'.

Similarly, a signature scheme is said to be $(t,\ \epsilon,\ q)$*-secure against existential forgery under an adaptive chosen message attack* if every t-time probabilistic algorithm B wins a modified game with probability less than ϵ. The game is identical to the above, except that the target message M is output by B in the last step (Step 4), at the same time as the candidate signature S. See Figure 2.

3 Equivalence of MMAC and Signing for Selective Forgery

One can easily show that for each notion of security defined above, every $(t,\ \epsilon,\ q)$-secure signature scheme is also a $(t,\ \epsilon,\ q)$-secure multicast authentication code. Our goal in the next two sections is to show an approximate converse: any short MMAC gives rise to a signature scheme with an almost equal level of security. We begin by showing that a MMAC secure against selective forgery gives rise to a signature scheme secure against selective forgery. In the next section, we show a similar result for existential forgery.

The derived signature scheme: Given a MMAC (key-gen, mac-gen, mac-ver) we define the derived signature scheme (skey-gen, sig-gen, sig-ver) as follows:

$\mathsf{skey\text{-}gen}(k, n)$	1. Run $\mathsf{key\text{-}gen}(k, n)$ to get $(\mathbf{sk}, \mathbf{rk}_1, \ldots, \mathbf{rk}_n)$. 2. Pick a random subset $I = \{i_1, \ldots, i_w\} \subseteq \{1, \ldots, n\}$. 3. Output $K_{\text{sec}} = \mathbf{sk}$ and $K_{\text{pub}} = (\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$.
$\mathsf{sig\text{-}gen}(M, K_{\text{sec}})$	Output $T = \mathsf{mac\text{-}gen}(M, K_{\text{sec}})$.
$\mathsf{sig\text{-}ver}(M, S, K_{\text{pub}})$	Write $K_{\text{pub}} = (\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$. Output 'yes' if and only if for all $j = 1, \ldots, w$, $\mathsf{mac\text{-}ver}(M, S, \mathbf{rk}_{i_j}) =$ 'yes'.

The following theorem shows that the derived signature scheme has nearly identical security properties as the MMAC.

Theorem 3.1 *Suppose the MMAC* (key-gen, mac-gen, mac-ver) *is* (t, ϵ, q)*-secure against selective forgery under an adaptive chosen message attack, and suppose the length of the output of* mac-gen$(M$, $\mathbf{sk})$ *is bounded above by* $\tau = n - m$ *for all* M *and* $\mathbf{sk}$. *Then the derived signature scheme* (skey-gen, sig-gen, sig-ver) *is* $(t, \epsilon + \frac{1}{2^m}, q)$*-secure against selective forgery under an adaptive chosen message attack.*

Note that taking $m = 80$ already results in a sufficiently secure signature scheme. Hence, whenever the MMAC length is slightly shorter than the number of receivers, n, the MMAC is easily converted into a secure signature scheme.

Proof Suppose we have a forger B that produces successful selective forgeries for the derived signature scheme (skey-gen, sig-gen, sig-ver). We build a forger A for the MMAC (key-gen, mac-gen, mac-ver). The proof will follow by contradiction. Recall that we model security as the probability of winning a game against a certain challenger. We describe how the algorithm A interacts with the challenger in this game, using B as a subroutine. See Figure 3.

Step 1: The algorithm A runs B to obtain the selected message M, which it forwards to the challenger as the message intended for its own selective forgery.

Step 2: Algorithm A chooses a random subset $I = \{i_1, \ldots, i_w\} \subseteq \{1, \ldots, n\}$ and sends this to the challenger. The challenger responds with $(\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$ for some $(\mathbf{sk}, \mathbf{rk}_1, \ldots, \mathbf{rk}_n)$ generated randomly by key-gen.

Step 3: The algorithm A sets $K_{\text{pub}} = (\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$ and sends K_{pub} to B. The distribution on K_{pub} is identical to the distribution on keys generated by skey-gen.

Step 4: Algorithm A now continues the execution of B, forwarding each query M_i to the challenger, and passing along each response T_i back to B. Note that T_i is a valid signature on M_i as defined by the derived signature scheme.

Step 5: After at most q queries, B outputs a signature forgery S for M. The algorithm A outputs S as its candidate MMAC forgery for M.

We show that A wins the selective forgery game for MMACs with probability at least ϵ. That is, S is a MMAC forgery with probability at least ϵ. The proof is based on the concept of a "bad pair". Let M' be a message in $\{0,1\}^*$ and let I' be a coalition $I' \subseteq \{1, \ldots, n\}$. We say that the pair (M', I') is *bad* if there is some tag $T \in \{0,1\}^{n-m}$ satisfying:

$$\forall i \in I' : \mathsf{mac\text{-}ver}(M', T, \mathbf{rk}_i) = \text{'yes'} \quad \text{and} \quad \forall j \notin I' : \mathsf{mac\text{-}ver}(M', T, \mathbf{rk}_j) = \text{'no'}.$$

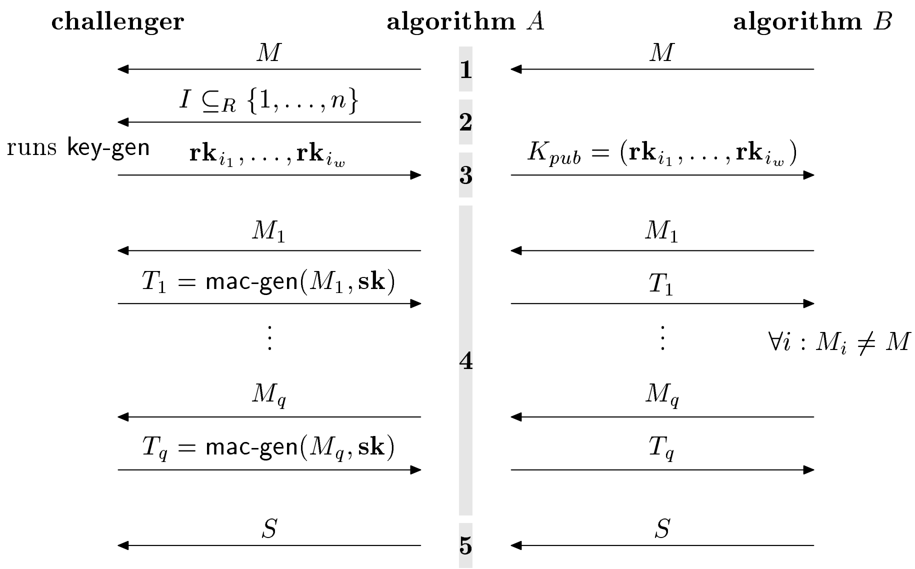

Figure 3: MMAC forger A uses signature forger B to forge a MMAC.

In other words, (M', I') is bad if I' is precisely the subset of receiver keys for which some tag T verifies as a valid tag for M'. The following lemma shows that for a fixed message M there are few pairs (M, I) that are bad.

Lemma 3.2 *For any message M:*

$$\Pr[(M, I) \textit{ is bad}] \leq \frac{1}{2^m}.$$

where the probability is over the choice of a random coalition $I \subseteq \{1, \ldots, n\}$.

Proof For each tag $T \in \{0,1\}^{n-m}$, let I_T be the set of receivers i for which $\mathsf{mac\text{-}ver}(M, T, \mathbf{rk}_i) =$ 'yes'. By definition, the pair (M, I_T) is bad. Notice that the collection

$$\left\{(M, I_T) \mid T \in \{0,1\}^{n-m}\right\}$$

completely describes all bad pairs containing M in the first coordinate. Since there are only 2^{n-m} possible values for T, this set is of size at most 2^{n-m}. Since I is chosen independently of M, it follows that

$$\Pr_{I \subseteq \{1,\ldots,n\}}[(M, I) \text{ is bad}] \leq \frac{2^{n-m}}{2^n} = \frac{1}{2^m},$$

establishing the lemma. ∎

We are now ready to complete the proof Theorem 3.1.

Proof of Theorem 3.1. We will establish the contrapositive. Suppose there is a forger B for the derived signature scheme (skey-gen, sig-gen, sig-ver) that runs in time t, makes q queries, and produces a successful selective forgery with probability at least $\epsilon + \frac{1}{2^m}$. We show the algorithm A described in Figure 3 wins the selective forgery game for the MMAC (key-gen, mac-gen, mac-ver) with probability at least ϵ.

We say that event $\mathcal{A}$ occurs when the pair (M, I) is **not** bad where M is the message chosen in Step 1, and I is the random set chosen in Step 2. We say the event $\mathcal{B}$ occurs when the algorithm B

wins the signature forgery game by outputting a forgery S on M in the derived signature scheme. By assumption we know that $\Pr[\mathcal{B}] > \epsilon + \frac{1}{2^m}$. Now, when both events $\mathcal{A}$ and $\mathcal{B}$ occur, we deduce the following:

(1) Since S is a signature forgery for M we have that

$$\forall i \in I : \quad \textsf{mac-ver}(M, S, \mathbf{rk}_i) = \text{`yes'};$$

(2) Since (M, I) is not bad, the set of users for which S is a valid MMAC cannot be I. Hence, by (1),

$$\exists j \notin I : \quad \textsf{mac-ver}(M, S, \mathbf{rk}_j) = \text{`yes'}.$$

But the second condition is precisely what is needed for A to win the selective forgery game against the MMAC. Since by Lemma 3.2 we have that $\Pr[\neg\mathcal{A}] \le \frac{1}{2^m}$ we obtain the following:

$$\begin{aligned} \Pr[A \text{ wins MMAC forgery game}] &\geq \Pr[\mathcal{B} \wedge \mathcal{A}] \geq \Pr[\mathcal{B}] - \Pr[\neg\mathcal{A}] \\ &\geq \left(\epsilon + \frac{1}{2^m}\right) - \frac{1}{2^m} = \epsilon. \end{aligned}$$

This probability is taken over the random coin flips of the challenger and of the algorithms A and B. Thus, the theorem follows. ∎

4 Equivalence of MMAC and Signing for Existential Forgery

Next, we show that an existentially secure MMAC gives rise to an existentially secure signature scheme. The resulting bounds are a bit weaker than for selective forgery. Let (key-gen, mac-gen, mac-ver) be a MMAC, and let H be a collision-resistant hash function from $\{0,1\}^*$ to $\{0,1\}^h$. Define the derived signature scheme (skey-gen, sig-gen, sig-ver) as follows:

$\textsf{skey-gen}(k, n)$	1. Run $\textsf{key-gen}(k, n)$ to get $(\mathbf{sk}, \mathbf{rk}_1, \ldots, \mathbf{rk}_n)$. 2. Pick random subset $I = \{i_1, \ldots, i_w\} \subseteq \{1, \ldots, n\}$. 3. Output $K_{\text{sec}} = \mathbf{sk}$ and $K_{\text{pub}} = (\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$.
$\textsf{sig-gen}(M, K_{\text{sec}})$	Output $T = \textsf{mac-gen}(H(M), K_{\text{sec}})$.
$\textsf{sig-ver}(M, S, K_{\text{pub}})$	Write $K_{\text{pub}} = (\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$. Output 'yes' if and only if for all $j = 1, \ldots, w$, $\textsf{mac-ver}(H(M), S, \mathbf{rk}_{i_j}) = \text{`yes'}$.

Suppose the MMAC (key-gen, mac-gen, mac-ver) is (t, ϵ, q)-secure against existential forgery under an adaptive chosen message attack, and suppose the length of the output of $\textsf{mac-gen}(M)$ is bounded by $\tau = n - m$ for all M. Furthermore let H be chosen from a family of collision-resistant hash function. specifically, suppose no t-time algorithm can find $M_1 \neq M_2$ such that $H(M_1) = H(M_2)$ with success probability greater than some small ϵ_H. We show in the following theorem that the derived signature scheme retains nearly identical security properties.

Theorem 4.1 *The derived signature scheme* (skey-gen, sig-gen, sig-ver) *is* $(t, \epsilon + \frac{1}{2^{m-h}} + \epsilon_H, q)$*-secure against existential forgery under an adaptive chosen message attack.*

For example, suppose (key-gen, mac-gen, mac-ver) is (t, ϵ, q)-secure against existential forgery. Let H be the hash function SHA-1 : $\{0,1\}^* \to \{0,1\}^{160}$, with security $\epsilon_H \approx \frac{1}{2^{80}}$. Then taking $m = 240$

results in a sufficiently secure signature scheme. Hence, as soon as the MMAC length is slightly less than the number of receivers, n, we obtain an existentially secure signature scheme.

Proof of Theorem 4.1. We will establish the contrapositive. Suppose we have a forger B that produces successful existential forgeries for the derived signature scheme ($\mathsf{skey\text{-}gen}$, $\mathsf{sig\text{-}gen}$, $\mathsf{sig\text{-}ver}$). We build a MMAC forger A for ($\mathsf{key\text{-}gen}$, $\mathsf{mac\text{-}gen}$, $\mathsf{mac\text{-}ver}$). Recall that we model security as the probability of winning a game against a certain challenger. We describe how the algorithm A interacts with the challenger in this game, using B as a subroutine.

Step 1: The algorithm A chooses a random subset $I = \{i_1, \ldots, i_w\} \subseteq \{1, \ldots, n\}$ and sends this to the challenger, which responds with $(\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$ for some $(\mathbf{sk}, \mathbf{rk}_1, \ldots, \mathbf{rk}_n)$ generated randomly by $\mathsf{key\text{-}gen}$.

Step 2: Algorithm A sets $K_{\text{pub}} = (\mathbf{rk}_{i_1}, \ldots, \mathbf{rk}_{i_w})$ and sends K_{pub} to B.

Step 3: For each query M_i made by B, algorithm A sends the query $H(M_i)$ to the challenger. Algorithm A then passes the response T_i back to B.

Step 4: After at most q queries, B outputs a message M and a candidate signature forgery S for M. If $H(M_i) = H(M)$ for some $i \in \{1, \ldots, q\}$, the algorithm A aborts the forgery attempt, as a collision in H has been found. Otherwise, the algorithm A outputs the pair $(H(M), S)$ as its candidate MMAC forgery.

We claim that A wins the existential forgery game for MMACs with probability at least ϵ. The proof uses the following concept: we say that a subset of users $I' \subseteq \{1, \ldots, n\}$ is *bad* if there is some $H_m \in \{0,1\}^h$ and some tag $T \in \{0,1\}^{n-m}$ such that

$$\forall i \in I' : \mathsf{mac\text{-}ver}(H_m, T, \mathbf{rk}_i) = \text{`}\mathsf{yes}\text{'}, \quad \text{and}$$
$$\forall j \notin I' : \mathsf{mac\text{-}ver}(H_m, T, \mathbf{rk}_j) = \text{`}\mathsf{no}\text{'}.$$

That is, I' is bad when I' is precisely the subset of receiver keys for which some tag T verifies as a valid tag for some H_m in the range of the hash function H.

Lemma 4.2 *When I is a random subset of $\{1, \ldots, n\}$ we have that:*

$$\Pr[I \textit{ is bad}] \leq \frac{1}{2^{m-h}}.$$

Proof We use the bound of Lemma 3.2 on the probability that a *pair* (H_m, I') is bad, for any $H_m \in \{0,1\}^h$. We obtain the following:

$$\begin{aligned}\Pr_{I \subseteq \{1,\ldots,n\}}[I \text{ is bad}] &= \Pr_{I \subseteq \{1,\ldots,n\}}[\exists H_m \in \{0,1\}^h \quad \text{s.t.} \quad (H_m, I) \text{ is bad}] \\ &\leq \sum_{H_m \in \{0,1\}^h} \Pr_{I \subseteq \{1,\ldots,n\}}[(H_m, I) \text{ is bad}] \leq 2^h \left(\frac{1}{2^m}\right) = \frac{1}{2^{m-h}},\end{aligned}$$

as desired. ∎

We can now complete the proof Theorem 4.1. Suppose there is a forger B for the derived signature scheme ($\mathsf{skey\text{-}gen}$, $\mathsf{sig\text{-}gen}$, $\mathsf{sig\text{-}ver}$) that runs in time t, makes q queries, and produces a successful existential forgery with probability at least $\epsilon + \frac{1}{2^{m-h}} + \epsilon_H$. We claim algorithm A described above wins the existential forgery game for the MMAC (key-gen, mac-gen, mac-ver) with probability at least ϵ.

We say the event $\mathcal{A}$ occurs when the set I chosen in Step 1 of algorithm A is **not** bad. We say the event $\mathcal{B}$ occurs when the algorithm A does **not** abort in Step 4. Finally, we say the event $\mathcal{C}$ occurs when the algorithm B wins the existential forgery game by outputting a forgery S on M in the derived signature scheme. By assumption we know that $\Pr[\mathcal{C}] \geq \epsilon + \frac{1}{2^{m-h}} + \epsilon_H$.

Now, when events $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ hold, we deduce the following:

(1)	$\forall i \in I:$	$\textsf{mac-ver}(H(M), S, \mathbf{rk}_i) = \text{'yes'}$	(S is a signature forgery for M),
(2)	$\forall i \in I:$	$H(M) \neq H(M_i)$	(A does not abort),
(3)	$\exists j \notin I:$	$\textsf{mac-ver}(H(M), S, \mathbf{rk}_j) = \text{'yes'}$	(by (1) and the fact that I is not bad).

But the second and third conditions are precisely what is needed for A to win the existential forgery game against a MMAC. So, by Lemma 4.2 and the fact that H is collision-resistant:

$$\begin{aligned}
\Pr[A \text{ wins MMAC forgery game}] &\geq \Pr[\mathcal{C} \wedge \mathcal{A} \wedge \mathcal{B}] \\
&\geq \Pr[\mathcal{C}] - \Pr[\neg\mathcal{A}] - \Pr[\neg\mathcal{B}] \\
&\geq \left(\epsilon + \frac{1}{2^{m-h}} + \epsilon_H\right) - \frac{1}{2^{m-h}} - \epsilon_H = \epsilon.
\end{aligned}$$

This probability is taken over the random coin flips of the challenger and of the algorithms A and B. Thus, the theorem follows. ∎

Note that the construction of the signature scheme above made use of a collision resistant hash function. The proof can be easily modified to only use one way universal hashing (OWUHF). Since OWUHF's can be constructed from one-way functions, there is no need to rely on collision resistance.

5 Coalitions of Limited Size

A MMAC (key-gen, mac-gen, mac-ver) is said to be (t, ϵ, q, κ)*-secure against selective forgery under an adaptive chosen message attack* if every t-time probabilistic algorithm A wins the game in Section 2 (depicted in Fig. 1) with probability less than ϵ, where the coalition I is subject to the constraint $|I| \leq \kappa$. Similarly, (t, ϵ, q, κ)-security against existential forgery is defined as (t, ϵ, q)-security against existential forgery where the coalition size $|I|$ is limited by κ. Note that for $\kappa = n$, these notions are exactly the same as those defined in Section 2; when $\kappa < n$, the security requirements are strictly weaker.

We show in this section that a (t, ϵ, q, κ)-secure MMAC with output length less than

$$\log_2 \sum_{i=0}^{\kappa} \binom{n}{i}$$

gives rise to a signature scheme of nearly equivalent security.

Let (key-gen, mac-gen, mac-ver) be a MMAC that is (t, ϵ, q, κ)-secure against selective forgery under an adaptive chosen message attack. Define the derived signature scheme (skey-gen, sig-gen, sig-ver) as in Section 3, with the modification that $\textsf{skey-gen}(s, n)$ picks a random subset $I \subseteq \{1, \ldots, n\}$ subject to the constraint $|I| \leq \kappa$.

Suppose the length of the output of mac-gen(M) is bounded by

$$\tau := \left\lceil \log \sum_{i=0}^{\kappa} \binom{n}{i} \right\rceil - m$$

for all M. Then we show:

Theorem 5.1 *The derived signature scheme* (skey-gen, sig-gen, sig-ver) *is* $(t,\ \epsilon + \frac{1}{2^m},\ q)$*-secure against selective forgery under an adaptive chosen message attack.*

The proof follows that of Theorem 3.1. Because of the restriction on the size of the coalition I, the following alternative to Lemma 3.2 is required.

Lemma 5.2 *For any fixed message* M*,*

$$\Pr[(M, I) \textit{ is bad}] \leq \frac{1}{2^m}.$$

where the probability is over the choice of a random coalition $I \subseteq \{1, \ldots, n\}$ *satisfying* $|I| < \kappa$*.*

Proof For each tag $T \in \{0,1\}^\tau$, there is exactly one set I_T containing precisely those receivers i for which mac-ver$(M, T, \mathbf{rk}_i)$ = 'yes'. By definition, the pair (M, I_T) is bad. The collection

$$\{(M, I_T) \mid T \in \{0,1\}^\tau\}$$

completely describes all bad pairs containing M in the first coordinate. Since there are only 2^τ possible values for T, this set is of size at most

$$2^\tau = 2^{-m} \sum_{i=0}^{\kappa} \binom{n}{i}.$$

Since I is chosen independently of M, it follows that

$$\Pr_{\substack{I \subseteq \{1,\ldots,n\} \\ |I| \leq \kappa}} [(M, I) \text{ is bad}] \leq \frac{2^{-m} \sum_{i=0}^{\kappa} \binom{n}{i}}{\sum_{i=0}^{\kappa} \binom{n}{i}} = \frac{1}{2^m},$$

establishing the lemma. ■

With this lemma in place, Theorem 5.1 follows just as Theorem 3.1.

An analogous theorem may be shown for security against existential forgery. Let (key-gen, mac-gen, mac-ver) be a MMAC that is $(t,\ \epsilon,\ q,\ \kappa)$-secure against existential forgery under an adaptive chosen message attack. Define the derived signature scheme (skey-gen, sig-gen, sig-ver) as in Section 3, with the modification that skey-gen(k, n) picks a random subset $I \subseteq \{1, \ldots, n\}$ subject to the constraint $|I| \leq \kappa$.

Suppose the MMAC (key-gen, mac-gen, mac-ver) is $(t,\ \epsilon,\ q)$-secure against existential forgery under an adaptive chosen message attack, and suppose the length of the output of mac-gen(M) is bounded by $\tau = (\log \sum_{i=0}^{\kappa} \binom{n}{i}) - m$ for all M. Furthermore assume that H is a collision-resistant hash function with security parameter ϵ_H. Then one can show

Theorem 5.3 *The derived signature scheme* (skey-gen, sig-gen, sig-ver) *is* $(t,\ \epsilon + \frac{1}{2^{m-h}} + \epsilon_H,\ q)$*-secure against selective forgery under an adaptive chosen message attack.*

The proof is similar to the proof of Theorem 4.1 with the appropriate modification to Lemma 4.2.

6 Conclusions

We gave precise definitions for Multicast MACs (MMACs) secure against selective and existential forgeries. Our main results show that a short collusion-resistant multicast MAC can be easily converted into a signature scheme. This shows a gap between the cryptographic resources needed for two party MACs (where signatures are not needed) and the resources needed for Multicast MACs. Our bounds justify the recent effort into designing signature schemes for a multicast environment [5, 12, 10, 7, 12]. Such schemes require minimal buffering on the sender's side and resist packet loss. We also note the constructions of [10, 11] that provide a short MMAC without non-repudiation by using some weak timing assumptions.

For small values of κ, our lower bound for κ-secure MMACs asymptotically matches the upper bound construction of Canetti et al. [1]. Hence, the Canetti et al. construction has optimal length (up to a small constant factor) for a MMAC that is based purely on pseudorandom functions.

References

[1] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast Security: A taxonomy and some efficient constructions", in IEEE INFOCOM '99, vol. 2, pp. 708–716, 1999.

[2] A. De Santis and M. Yung, "On the design of provably-secure cryptographic hash functions", in Proc. of Eurocrypt '90, LNCS 473, pp. 412–431, 1990.

[3] Y. Desmedt, Y. Frankel, and M. Yung, "Multi-receiver/Multi-sender network security: efficient authenticated multicast/feedback", in IEEE INFOCOM '92, pp. 2045–2054, 1992.

[4] F. Fujii, W. Kachen, and K. Kurosawa, "Combinatorial bounds and design of broadcast authentication", in IEICE Trans., vol. E79-A, no. 4, pp. 502–506, 1996.

[5] R. Gennaro and P. Rohatgi, "How to sign digital streams", in Proc. of Crypto '97, 1997.

[6] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks", SIAM Journal of Computing, vol. 17, pp. 281–308, 1988.

[7] P. Golle, N. Modadugo, "Streamed authentication in the presence of random packet loss", in Proc. of 8th Annual Internet Society Symposium on Network and Distributed System Security (NDSS '01), San Diego, 2001.

[8] K. Kurosawa, S. Obana, "Characterization of (k, n) multi-receiver authentication", in Information Security and Privacy, ACISP '97, LNCS 1270, pp. 205–215, 1997.

[9] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications", in Proc. of 21st Annual ACM Symposium on Theory of Computing, pp. 33–43, 1989.

[10] A. Perrig, R. Canetti, D. Tygar, D. Song, "Efficient Authentication and Signature of Multicast Streams over Lossy Channels", in Proc. of 2000 IEEE Symposium on Security and Privacy, Oakland, 2000.

[11] A. Perrig, R. Canetti, D. Song, D. Tygar, "Efficient and Secure Source Authentication for Multicast", in Proc. of 8th Annual Internet Society Symposium on Network and Distributed System Security (NDSS '01), San Diego, 2001.

[12] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet authentication", in Proc. of 6th ACM conference on Computer and Communication Security, 1999.

[13] J. Rompel, "One-way functions are necessary and sufficient for secure signatures", in Proc. of 22nd Annual ACM Symposium on Theory of Computing, pp. 387–394, 1990.

[14] R. Safavi-Naini, H. Wang, "Multireceiver authentication codes: models, bounds, constructions and extensions", Information and Computation, vol. 151, no. 1/2, pp. 148–172, 1999.

[15] R. Safavi-Naini, H. Wang, "New results on multireceiver authentication codes", in Proc. of Eurocrypt '98, LNCS 1403, pp. 527–541, 1998.

[16] G. Simmons, "A cartesian product construction for unconditionally secure authentication codes that permit arbitration", *J. Cryptology*, vol. 2, no. 2, pp. 77–104, 1990.

[17] C. K. Wong, S. S. Lam, "Digital signatures for flows and multicasts", IEEE ICNP '98. Also, University of Texas at Austin, Computer Science Technical report TR 98-15.