Anonymous Authentication with Shared Secrets

Joël Alwen[1], Martin Hirt[1], Ueli Maurer[1], Arpita Patra[2], and Pavel Raykov[1]

[1] Department of Computer Science, ETH Zurich, Switzerland.
{alwenj,martin.hirt,ueli.maurer,pavel.raykov}@inf.ethz.ch
[2] Applied Statistics Unit, ISI Kolkata, India
arpitapatra10@gmail.com

Abstract. Anonymity and authenticity are both important yet often conflicting security goals in a wide range of applications. On the one hand for many applications (say for access control) it is crucial to be able to verify the identity of a given legitimate party (a.k.a. entity authentication). Alternatively an application might require that no one but a party can communicate on its behalf (a.k.a. message authentication). Yet, on the other hand privacy concerns also dictate that anonymity of a legitimate party should be preserved; that is no information concerning the identity of parties should be leaked to an outside entity eavesdropping on the communication. This conflict becomes even more acute when considering anonymity with respect to an active entity that may attempt to impersonate other parties in the system.
In this work we resolve this conflict in two steps. First we formalize what it means for a system to provide both authenticity and anonymity even in the presence of an active man-in-the-middle adversary for various specific applications such as message and entity authentication using the constructive cryptography framework of [Mau11]. Our approach inherits the composability statement of constructive cryptography and can therefore be directly used in any higher-level context. Next we demonstrate several simple protocols for realizing these systems, at times relying on a new type of (probabilistic) Message Authentication Code (MAC) called *key indistinguishable* (KI) MACs. Similar to the key hiding encryption schemes of [BBDP01] they guarantee that tags leak no discernible information about the keys used to generate them.

1 Introduction

1.1 Anonymous Authentication

Anonymity and authenticity are both important yet often conflicting security goals in a wide range of applications. On the one hand "entity authentication" is a core functionality needed for implementing access control both in physical and digital systems. Moreover for many applications we are also required to authenticate *what* is being said, a security goal more commonly referred to as "message authentication". In both cases an implicit assumption underlying the systems is that each user has some unique identifying information associated

with them which they can use either to prove who they are or what they are saying.

On the other hand, in a world where privacy matters, providing identifying information over public channels leads to an inherent conflict between the desire for authenticity (the property that no one else can claim to be you) and anonymity (the guarantee that external parties learn nothing about your identity). The problem is especially acute in light of the fact that many authentication protocols for physical access control are implemented using RFID tokens, where the communication can easily be eavesdropped, and the tokens can often be accessed wirelessly from a significant distance and without the consent or even awareness of the owner. Moreover mobile phones, constantly communicating over the public radio spectrum, also make use of uniquely identifying information for authenticating their communication with the network. Even on the internet when using a proxy or onion routing service to hide ones IP address a user interacts with a service requiring some form of authentication (say a VPN) may still not enjoy anonymity if the service does not make use of an anonymous authentication protocols.

In particular we stress that using cryptographic tools (to achieve secrecy and/or authenticity) over an anonymous channel generally destroys the anonymity of the channel. For example, if a challenge-response protocol based on a MAC and shared secret keys is used for client authentication, then the MAC values (aka tags) may leak partial information about the key, which means that an adversary can recognize that the same client is involved in different sessions, i.e., one loses unlinkability and hence also anonymity.

The goal of this work is to resolve this conflict allowing for the design of systems which provably guarantee both properties regardless of the greater context in which they are used.

1.2 Our Contributions

On the highest level we achieve our stated goal via two phases. First we cleanly, formally and composably capture what it means for a system to provide both authenticity and anonymity even in the presence of an active man-in-the-middle adversary for an array of specific applications such as message and entity authentication. Next we prove the security of several simple protocols for realizing these systems, at times relying on a new type of Message Authentication Code (MAC) introduced in [AHM+14].

Formalizing Anonymous Authentication. In more detail, the first contribution of this work is to intuitively model a variety of resources providing anonymity. We do this in the constructive cryptography framework of [Mau11][3], inheriting its general composability guarantees. Concretely we define anonymous variants of insecure channels ($\mathcal{F}_{A\text{-}IC}$), authenticated channels ($\mathcal{F}_{A\text{-}AC}$), secure channels ($\mathcal{F}_{A\text{-}SC}$) and entity authentication ($\mathcal{F}_{A\text{-}EA}$). Each primitive is modelled as an

[3] One could also give an equivalent formulation in the UC framework.

ideal resource which explicitly shows the abilities and limits of an active man-in-the-middle adversary. For example, the ideal resource of an anonymous secure channel allows a sender to send a message to a receiver such that the adversary learns only the length of the message. The only actions permitted to the adversary are to cause delivery of any message previously sent by a sender (but without learning either the contents of the message or the identity of its sender).

Constructions. The second contribution is to prove security for various constructions of stronger anonymous primitives from weaker ones (see Figure 1 for the overview). In particular, we build anonymous variants of authenticated channels from insecure channels (and a pairwise shared-key setup denoted with $\mathcal{K}$), entity authentication from authenticated channels (and a type of insecure broadcast channel denoted with $\mathcal{F}_{IB}$) and secure channels from authenticated channels. While most of the constructions are relatively immediate the proofs are often significantly more involved and it is these we consider to be the second contribution of this work. Some of the information theoretic constructions are decidedly unpractical (and should be viewed more as feasibility results) but we also provide several optimizations decreasing both communication and computational complexity. Combined with the pseudo-random function (PRF) based MAC of [AHM+14] these give rise to practically interesting protocols for constructing anonymous message authentication and entity authentication from anonymous insecure channels and insecure broadcast in the shared key setting.

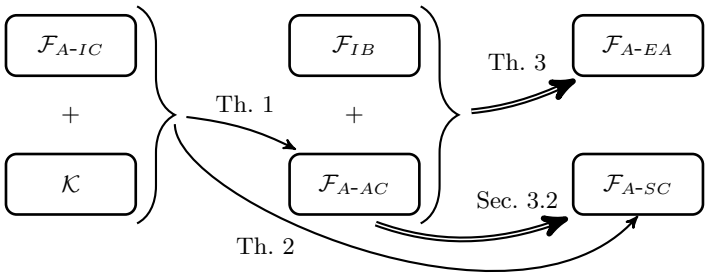


Fig. 1. The overview of resources and relations among them. The single arrow ($\longrightarrow$) denotes computational constructions, while double arrow ($\Longrightarrow$) denotes information-theoretic constructions.

All computational constructions in this work rely on a novel primitive called *key indistinguishable* (KI) message authentication codes (MAC). Similar to the notion of a key hiding public key encryption scheme [BBDP01] these are (probabilistic) MACs such that tags generated with *different* keys cannot be distinguished from tags generated with the *same* key.[4] This notion was introduced in [AHM+14] where a variety of constructions are given based on both black-box primitives (such as PRFs, weak PRFs and HPS) as well as concrete number theoretic assumptions (such as DDH, LWE and LPN).

[4] Or more generally using the same or different states.

Determnistic Devices. We remark that while most of our protocols require parties to be probabilistic (which may be a problem for extremely light-weight computing devices) they can easily and generically be translated into stateful but deterministic parties by using a PRG.[5]

Robustness to Side-Channels. An important consequence of how we define these resources is that we capture various types of information which may potentially be available to an adversary through side-channels. Technically this is done by providing extra capabilities and information to the distinguisher[6] D whose goal it is to tell the construction and ideal resource apart. In particular while no information concerning the identities of anonymous parties leaks on the adversarial interface of a given resource, D can trivially obtain such information from say the receivers interface. Thus D can make use of such information to aid in its task. In other words even an adversary equipped with such a side-channel learns no more from its interface to the real communication resource used to run the protocol than it does from the side-channel and the adversarial interface to the the ideal resource. It is this property which is central to the intuitive claim of composability for all our constructions.[7]

Besides side-channels leaking information related to identities another important type of side-channel we model concerns the relative order in which parties respond. In particular responses are always initiated at the behest of D. This is of particular interest in a setting with mobile phones of differing computational power or RFID tags positioned at differing distances from the adversary.

Exact Security. All reductions we give come with an exact security analysis (as opposed to asymptotic ones). We see at least two advantages in taking this approach. First, such results greatly facilitate comparing the quality/efficiency trade-off obtained via different constructions especially when based on the same underlying cryptographic assumptions. A somewhat less common but equally relevant advantage is that such statements make explicit the benefits obtained by enforcing constraints on the adversary through implementation choices. Take for example a protocol whose security degrades say in $q/\left|\mathcal{M}\right|$: the number of times an adversary can interact with a client divided by the size of the messages space supported by a MAC. Normally such a protocol would require a MAC with at least 160-bit messages to be considered secure. However, if implemented on hardware which guarantees failure after a limited number of interactions, say $q \leq 2^{10}$ (a common assumption in the RFID setting) the MAC now needs only to support 100-bit messages potentially reducing the hardware costs of the resulting implementation significantly.

[5] In particular the security proof for the probabilistic setting then automatically carries over (at least in a computational sense) by preceding the proof with a hybrid argument replacing the output of each call to the PRG with fresh random numbers.

[6] called the "environment" in the language of UC.

[7] And more abstractly, this property plays an important role in the composition theorem of [Mau11].

1.3 Related Work

We provide an overview of the related literature which can, roughly speaking, be divided up into those concerned with generic entity authentication, anonymous entity authentication for RFIDs and anonymous authentication for mobile phone networks.

Entity Authentication. A large body of work going back almost 30 years has considered a range of security notions for unilateral entity authentication which, broadly speaking, consist of an "information collection" phase followed by an "attack" phase. An elegant overview elucidating their relationships can be found in [MT12]. Borrowing the language of [MT12] we can informally characterise a given security notion using three types of oracles C, S and T, namely the honest client (prover), server (verifier) and a transcript oracle[8] respectively. A particular security notion is then defined via two subsets of these oracles indicating the resources available to the adversary during the two respective phases. For example $(\{C\},\{S\})$-auth security, traditionally referred to as *active* security, allows the adversary oracle access to the client in the first phase while only allowing access to the server in the attack phase. Another common example is $(\{T\},\{S\})$-auth which is traditionally referred to as *passive* security.

While classical entity authentication protocols [FS86, GQ88, Sch89, Oka92] focus on the public key setting satisfying weaker security notions (i.e. *passive* and *active* security) more recent works (especially in the context of RFID identification protocols) are set in the shared-key model and achieve increasingly strong types of security culminating in several variants of man-in-the-middle (MiM) security. In this work we consider $(\{C,S\},\{S\})$-auth which [MT12] show to be strictly weaker than $(\{\},\{C,S\})$-auth (also used in [BR93, Vau10] for example). We justify this choice by arguing that the later MiM variant is in fact stronger than needed in real world applications. In particular it is unavoidable that an adversary, with online access to C during the attack phase, can convince S to accept (e.g. by blindly forwarding messages). Thus, in contrast to $(\{\},\{C,S\})$-auth, we have opted for a security notion which does not rule out adversaries convincing S using an *online* C through more involved means such as by modifying C's messages [9].

Anonymous Public-Key Encryption. Kohlweiss et al. [KMO+13] initiated the study of anonymity in the constructive cryptography framework. They considered a single sender multi-receiver setting where the sender and receivers communicate via a receiver-anonymous insecure channel. In such a setting they apply a public-key encryption in order to achieve a receiver-anonymous *confidential* channel. While achieving confidentiality with a public-key encryption is straightforward, one needs to additionally assume that the employed scheme is

[8] Upon each invocation the transcript oracle outputs a freshly sampled transcript between the honest server and client.

[9] As is done for example in the separating example between the two notions in [MT12].

key-private [BBDP01], i.e., which essentially means that for two given public keys and one ciphertext, one cannot decide for which public key this ciphertext is valid.

Anonymous Authentication for RFIDs. Anonymous authentication has primarily been studied in the context of entity authentication especially in the context of RFID systems, where anonymity is of particular interest. Unfortunately these results are tailored to this specific application and do not put forward a general framework for anonymous authentication as in this work.

Furthermore, most of the suggested schemes are proven secure using game-based notions [Vau10,HPVP11,DLYZ11], where security of the real world system is guaranteed only within the particular (often complex) context described by the game. Therefore it often remains unclear within which greater context such protocols can be used. For example, in perhaps the most popular such notion of [Vau10] and its derivatives (ex. [HPVP11]), the anonymity of a new session is defined only against adversaries which do not learn the identities of clients involved in previous (successful) sessions. In particular this means such protocols can not, a priori, be used in a greater context where an adversary may have access to say the full output of an RFID reader at any point in time. However ideally one would hope for a protocol which is again anonymous once the adversary looses such side-channel access to the card reader.

Composable security (UC) is considered in [ACdM05], but this solution assumes purely passive tags, which an adversary can easily overwrite, hence security cannot be achieved against a MiM adversary. Recently, [BLdMT09] and [BM11] provide composable (UC) security, but the clients must be stateful.

Note that the term "anonymous authentication" is at times also used for group authentication, where the *server* must not learn which particular client is authenticating. This is not the scope of this paper.

Anonymous Authentication for Mobile Phone Networks. Besides the RFID setting the models and definitions in this work also apply to authentication requirements in other radio communication networks such as mobile and satellite phone systems. For example, in the specification of the 3rd (and later) generation mobile phone systems by the 3GPP, a crucial key agreement (KA) phase between end-users and the network operator is described. While this phase involves communication over the public radio spectrum (an insecure, but potentially anonymous communication channel) the specification explicitly lists both end-user anonymity (preservation) and authenticity as key design goals for this phase [rGPP12]. To achieve this, a unique long-term secret is shared between each end-user device, called a Universal Subscriber Identification Module (USIM), and the services provider. Crucially, because the actual mobile devices (e.g. cellphones, tablets) are not trusted, all secret key operations on the user-end are performed on the USIM; a very light-weight computing device. This in turn has lead to the use of light-weight cryptographic primitives in the form of symmetric key algorithms such as those used in this work.

Existing solutions leave much to be desired both in terms of anonymity [KAC08, AMRR11, AMR+12] and authenticity [TM12] and a large body of work with ad-hoc security arguments focuses on improving the status quo (e.g. [BR05, KO05, GVI06, SAJ07, CRS11, CRS12]). Notable exceptions are the works of [AMRR11, TM12, AMR+12] which make use of more formal symbolic analysis to discover some vulnerabilities in existing solutions. Finally in the concurrent and independent work of [LSWW13] a complex game-based security model, targeted specifically at the setting of UMTS/LTE client-network communication (including the KA), is presented. Moreover existing solutions are shown to satisfy a limited notion of anonymity and (roughly speaking) $(\{C, S\}, \{S\})$-authenticity under some novel yet plausible assumptions concerning key component functions used in the protocols.

In light of these results we view our work as making significant progress towards developing a formal yet intuitively tractable and secure model, compatible with the language and tools of modern cryptography, for capturing and analyzing the stated design goals of mobile phone networks together with examples of rigorously analyzed solutions. While we by no means claim the model (nor protocols) to be directly applicable in this arena we do believe them to represent an important step forward towards developing more satisfactory security guarantees for this extensively deployed application.

1.4 Outline

In Section 2 we briefly review the constructive cryptography framework [Mau11] (with support for making exact security statements), and review two security notions for MAC schemes.

In Section 3 we present the anonymous authentication resources for various applications and provide information theoretic and computational constructions. We also describe some more efficient variants thereof. In particular we describe a protocol providing a trade-off allowing a potentially much more efficient server protocol for realizing entity authentication at the cost of maintaining a short but mutable state on the client side.

2 Definitions

We review the constructive cryptography framework which we use to define anonymous authentication protocols. Next, in this section we define several security notions for MACs including key-indistinguishability as well as a variety of unforgeability definitions.

2.1 Constructive Cryptography and Exact Security

The primary goal of this work is to build anonymous authentication protocols under various assumptions such that the resulting protocols can be used as a

building block within *any* greater system. This gives rise to two defining characteristics of our security notions.

On the one hand we require an *arbitrarily* composable security guarantee. For this we use the constructive cryptography [Mau11] (CC) framework which allows for real/ideal type definitions supporting very strong "general" composability. On the other hand in practice concrete security matters. So departing somewhat from the asymptotic statements used in many such definitions we provide precise security claims detailing the exact security as a function of the properties of the underlying assumptions. In doing so we provide a method to evaluate the practicality and efficiency of using different constructions and basing security on different concrete assumptions.

Recalling the CC Framework. We recall the main ideas of the CC framework and refer to [Mau11] for further details. Similar to the Universal Composability (UC) framework of Canetti [Can01] the CC framework makes use of two types of basic computational systems. The first are called *resources* which are equipped with a set of interfaces[10]. We generally denote resources using calligraphic capital letters such as $\mathcal{R}$ and $\mathcal{F}$. Each interface captures the capabilities of a particular party which interacts with the system. The second type of systems are called *converters* (such as a protocol π or simulator σ). Converters have an internal interface through which they are connected to the available resources and an external interface through which the composed system is accessed by the context in which it is being used. In contrast to the UC framework we do not assume the presence of a network of insecure channels and instead explicitly define all communication resources we use in any given statement.

The CC framework allows for the presence of several resources for which the $||$ operator is used. For example the system where say resources modeling a setup $\mathcal{R}$ and an insecure channel $\mathcal{F}$ are present is denoted by $(\mathcal{R}||\mathcal{F})$. Moreover resources can be composed with (possibly multiple) converters giving rise to a network of computational systems which we refer to as a *composed system*. More concretely a broadcast channel (resource) together with several protocols (converters) forms a composed system. To denote this composed system obtained by attaching say a protocol π to resource $\mathcal{R}$ on interface I we write $\pi^I\mathcal{R}$.[11] As usual a security definition in the CC framework involves equating two composed systems; intuitively modeling the "real" and "ideal" worlds respectively.

We wish to capture the intuition of providing "security against an attacker". For this we model the capabilities of an adversary using a given resource via an *adversarial interface*. Generally a resource $\mathcal{R}$ modeling the real world provides non-trivial capabilities on the adversarial interface (say full man-in-the-middle (MiM) access) while the ideal resource $\mathcal{F}$ only provides very restricted capabilities on that interface. The desired intuition is then captured by detailing a

[10] In the language of UC we speak of ideal functionalities and of ITM communication tapes in the language of ITMs.

[11] We note that resources and composed systems are actually computational objects of the same type and so at times we also use calligraphic capital letters to denote a composed system.

simulator converter which attaches to adversarial interface of $\mathcal{F}$ and produces an on-line translation (a.k.a. a simulation) making the interface look like the adversarial interface of $\mathcal{R}$. Unlike UC, the CC framework does not model the adversary as an (arbitrary) separate entity (converter). Instead CC's approach could be thought of as UC in the special case of the dummy adversary, which acts as a transparent conduit between its inner and outer interfaces.[12]

Finally the CC framework allows for security definitions which support "general composition". That is the real and ideal composed systems are interchangeable *regardless* of the context in which they are used.[13] For this CC like UC uses an online adaptive distinguisher D (i.e. the "environment" in UC) whose goal it is to tell the two systems apart. Intuitively D models the arbitrary context in which the systems might be used and relative to which they should be interchangeable. Technically D is given access to all interfaces of either the real or ideal system and, after arbitrary interaction with the system, D outputs a bit indicating whether it believes this was the real or the ideal system.

Exact Security. We develop the notation for making exact security statements in the CC framework. Take a fixed pair of systems $\mathcal{R}$ and $\mathcal{F}$ with k interfaces. We consider a parametrized class of $(t, x_1, \ldots, x_k)$-distinguishers D running in time t and querying the i^{th} interface at most x_i times during the interaction with a system ($\mathcal{R}$ or $\mathcal{F}$). The advantage of a specific D from this class $\Delta^{\mathsf{D}}(\mathcal{R}, \mathcal{F})$ is defined to be $|\Pr[\mathsf{D}(\mathcal{R}) \to 1] - \Pr[\mathsf{D}(\mathcal{F}) \to 1]|$. We will write $\mathcal{R} \approx_\alpha \mathcal{F}$ (where $\alpha = (t, x_1, \ldots, x_k, \epsilon)$) to say that all distinguishers from this class have advantage at most ϵ.[14] In this work all systems are also parameterized by a implicite security parameter λ. It can be understood to be fixed to an arbitrary value once and for all and then shared across all systems in any given theorem.

2.2 Message Authentication Codes (MAC)

In this subsection we define the syntax and several security properties for message authentication codes.

[12]Indeed, as shown in the so called "Dummy Lemma" for various UC type frameworks, this restriction results in no loss of generality while making security proofs far more tractable.

[13]This stands in contrast to say game based definitions which instead guarantee certain properties of a real world system only within the particular context captured by the game. For example the anonymity of the authentication protocols defined in [Vau10, HPVP11] holds only with respect to adversaries which remain oblivious to which parties have previously authenticated themselves during the life of the system (even for the "wide adversary" variants).

[14]More specifically in this work the underlying cryptographic assumptions used give rise to the properties of the real world resource $\mathcal{R}$ while the implementation choices can allow for bounding properties of D. The final distinguishing advantage of the real and ideal systems is usually a function of both types of properties.

Syntax. A message authentication code $\mathsf{MAC} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ is a triple of algorithms with associated key space $\mathcal{K}$, message space $\mathcal{M}$, and tag space $\mathcal{T}$.

- **Key Generation.** The probabilistic key generation algorithm $k \leftarrow \mathsf{KG}(1^\lambda)$ takes as input a security parameter $\lambda \in \mathbb{N}$ (in unary) and outputs a secret key $k \in \mathcal{K}$.
- **Tagging.** The probabilistic authentication algorithm $\tau \leftarrow \mathsf{TAG}_k(m)$ takes as input a secret key $k \in \mathcal{K}$ and a message $m \in \mathcal{M}$ and outputs an authentication tag $\tau \in \mathcal{T}$.
- **Verification.** The deterministic verification algorithm $\mathsf{VRFY}_k(m, \tau)$ takes as input a secret key $k \in \mathcal{K}$, a message $m \in \mathcal{M}$ and a tag $\tau \in \mathcal{T}$ and outputs an element of the set $\{\texttt{Accept}, \texttt{Reject}\}$.

Next we define some useful properties such a triple of algorithms can have such as completeness and unforgeability. We also discuss a less common security notion for MACs, called key indistinguishability [AHM+14] which can only be achieved by *randomized* MACs. Each of the following definitions depend on a security parameter λ. However, in line with the above discussion on the treatment of the security parameter in our constructive statements, we ommit λ from our notation. Instead, to avoid clutter, we assume that all security properties in any given statement share the same fixed value of λ.

Completeness. We say that MAC has completeness error η if for all $m \in \mathcal{M}$,

$$\Pr[\mathsf{VRFY}_k(m, \tau) = \texttt{Reject} : \quad k \leftarrow \mathsf{KG}(1^\lambda), \tau \leftarrow \mathsf{TAG}_k(m)] \leq \eta.$$

Unforgeability. We recall the standard notion security for (randomized) MACs; namely unforgeability under chosen message (and chosen verification) query attack (**uf-cmva**). We denote by $\mathbf{Adv}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda)$, the *advantage* of the adversary A in forging the message for a random key $k \leftarrow \mathsf{KG}(1^\lambda)$. Formally it is the probability that the following experiment outputs 1.

Experiment. $\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda)$
- $k \leftarrow \mathsf{KG}(1^\lambda)$
- *Invoke* $\mathsf{A}^{\mathsf{TAG}_k(\cdot), \mathsf{VRFY}_k(\cdot,\cdot)}$.
- *Output* 1 *if* A *queried* (m^*, τ^*) *to* $\mathsf{VRFY}_k(\cdot,\cdot)$ *s.t.* $\mathsf{VRFY}_k(m^*, \tau^*) = \texttt{Accept}$ *and* A *did not receive* τ^* *by querying* m^* *to* $\mathsf{TAG}_k(\cdot)$.

The above experiment can be weakened by relaxing point 2. in the winning condition of experiment $\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}$ to require that m^* has not previously been queried to $\mathsf{TAG}_k(\cdot)$. We refer to the resulting notion as *weakly* unforgeable while referring to the more stringent security notions as *strongly* unforgeable. In general in this work unless stated explicitly otherwise we always mean the strong variants. Finally we can remove the adversary's access to the verification oracle in which case we refer the the experiment as **cma** rather than **cmva**.

We refer to an efficient (i.e. PPT) adversary A playing a **cmva** type experiments as a (t, q_t, q_v)-adversary if it runs in time at most t, and for any pair of oracles with a fixed key A makes at most q_t tag and q_v verification queries.

Definition 1 (Unforgeability of MACs). *A message authentication scheme* MAC *is* (t, q_t, q_v, ϵ)-**uf-cmva** secure *if for any* (t, q_t, q_v)*-adversary* A *we have:*

$$\mathbf{Adv}^{\textsf{uf-cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda) := \Pr[\mathbf{Exp}^{\textsf{uf-cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda) \to 1] \leq \epsilon.$$

Key Indistinguishability. The notion of key indistinguishability (KI) guarantees that tags leak no information about the underlying key (or state). This allows us to use such a scheme to implement authentication anonymously. We note that such a property is not implied by even the strongest of unforgeability notions defined above.[15] The intuition we capture for KI is that an adversary can not tell a single pair of tag and verify oracles from two pairs of such oracles with different states (including secret keys). In other words if an adversary has access to 4 oracles (2 tag and 2 verify oracles) it can not tell if the tag (and verify) oracles actually use the same state or not.

To formalize this we introduce some notation. For keys $k_0, k_1 \in \mathcal{K}$ we write $[k_0, k_1]$ to denote the 4-tuple of oracles $(\mathsf{TAG}_{k_0}, \mathsf{VRFY}_{k_0}, \mathsf{TAG}_{k_1}, \mathsf{VRFY}_{k_1})$. Moreover we write $[k_0, k_0]$ to denote a similar 4-tuple but where the TAG oracles share their entire internal state including secret key (and similarly for the VRFY oracles). In other words calls to the first and third oracle of $[k_0, k_0]$ are answered by essentially the same oracle (and similarly for the second and fourth oracle).[16]

Experiment. $\mathbf{Exp}^{\textsf{ki-cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda)$
- $k_0, k_1 \leftarrow \mathsf{KG}(1^\lambda)$, $c \leftarrow \{0, 1\}$
- *Sample output* $c' \leftarrow \mathsf{A}^{[k_0, k_c]}$.
- *If a tag obtained from the left oracle (namely* TAG_{k_0}*) was verified using the right verification oracle (namely* VRFY_{k_c}*) or vice versa, then output a uniform random bit.*
- *Otherwise if* $c = c'$ *output* 1 *and* 0 *otherwise.*

As usual, in the above experiment we have made a non-triviality constraint; namely that A is not allowed to make a verification query (m, τ) to oracle VRFY_{k_c} if τ was obtained from TAG_{k_0} for message m (and vice versa).

As before in the following definition we say that an adversary A is a (t, q_t, q_v)-adversary if it runs in time at most t and for *each* pair of oracles with a given key

[15] Indeed this is not difficult to see. For example we can modify any (say **uf-cmva**) unforgeable scheme as follows such that it is clearly not key indistinguishable. Double the key size, use the first half of the key in conjunction with the original TAG algorithm to tag the message and then append the second half of the key to the resulting tag. Clearly the scheme remains unforgeable however it is trivial to tell tags issued under different keys apart.

[16] For stateful MACs it is important that the full state (and not just the secret key) be shared between matching oracles in $[k_0, k_0]$. Suppose we have a secure MAC which hides all information about the secret keys. We can modify the TAG algorithm to keep a counter which it appends to each tag τ it outputs. Clearly the scheme still hides all information about the secret key. However it is unclear how such a scheme might be used to achieve anonymity. Indeed it is trivial to tell say the 10^{th} tag issued for key k_0 from the 3^{rd} tag issued for different key k_1.

makes at most q_t tag and q_v verification queries. So in total such an adversary can make up to $2q_t$ tag queries namely by making q_t queries to TAG_{k_0} and TAG_{k_c}.

Definition 2 (Key Indistinguishability). *Let λ be an (implicit) fixed security parameter. A message authentication scheme* MAC *is* (t, q_t, q_v, ϵ)-**ki-cmva** secure *if for any (t, q_t, q_v)-adversary* A *we have:*

$$\mathbf{Adv}^{\textbf{ki-cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda) := 2\left|\Pr[\mathbf{Exp}^{\textbf{ki-cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda) \to 1] - \frac{1}{2}\right| \leq \epsilon$$

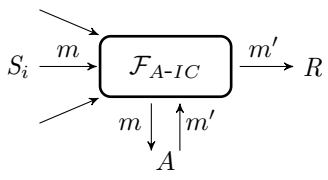
Moreover if MAC *is* $(t, q_t, 0, \epsilon)$-**ki-cmva** *then we call it* (t, q_t, ϵ)-**ki-cma** secure*. In particular in the* **ki-cma** *experiment we simply omit all verification oracles.*

3 Anonymous Authentication as Real/Ideal Transformations

In this section we define a range of anonymous resources together with various computational and information theoretic protocols for constructing them. We also describe several optimizations including two practically relevant protocols.

3.1 Anonymous Message Authentication

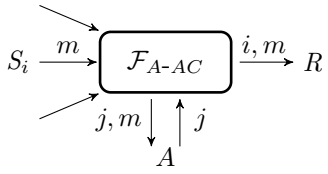
We begin by focusing on anonymous message authentication. We prove that using KI and unforgeable MAC one can construct anonymous variants of authenticated channels from insecure channels in the shared key setting thereby reducing the problem of anonymous message authentication to building such MACs. In Appendix A.3 we also give an optimization which provides the trade-off of improving receiver efficiency (given an optimistic but realistic assumption) at the cost of requiring senders to be stateful.[17]



We define an anonymous insecure channel $\mathcal{F}_{A\text{-}IC}$ (intuitively depicted on the right side) which captures the minimal communication resource we require for achieving any type of anonymous authentication. Intuitively this is a multi-sender/single-receiver channel which provides the guarantee that the identity of the sender remains hidden on the adversary's interface. The scheduling and content of messages being sent is externally driven (technically they are provided by the distinguisher D) and we model an active adversary with full control over message delivery and content. Finally once the adversary chooses to deliver a message to the receiver, the receiver learns the content of that message but not (a priori) the identity of the original sender. Indeed this identity may not even be well defined as the adversary may have mauled an original sent message or even invented a completely new message for delivery.

[17] For some applications (such as entity authentication for light-weight devices) this reflects a design choice for senders already common in practice.

Next we define a multi-sender/single-receiver anonymous *authenticated* channel $\mathcal{F}_{A\text{-}AC}$, (intuitively depicted on the right side and described formally in Figure 2). The difference to $\mathcal{F}_{A\text{-}IC}$ are twofold. First the adversary is now restricted to delivering only messages m which were original sent by one of the senders and second upon delivery of m the receiver additionally learns the identity of the original sender.



INIT: $M \leftarrow \emptyset$, *counter* $\leftarrow 0$
ON INTERFACE S_i:
CASE (m): *counter* $\leftarrow$ *counter* $+ 1$, $M \leftarrow M \cup \{(counter, i, m)\}$; output $(counter, m)$ on interface A
ON INTERFACE A:
CASE $(j \in \mathbb{N} \cup \{\perp\})$: If $\exists (j, i, m) \in M$ then output (i, m) on interface R (and otherwise output $\perp$).

Fig. 2. The anonymous authenticated channel $\mathcal{F}_{A\text{-}AC}$

Finally we construct $\mathcal{F}_{A\text{-}AC}$ from $\mathcal{F}_{A\text{-}IC}$ in a shared key setting modeled via n key-distribution resources $\mathcal{K} = \{\mathcal{K}_i \mid i \in [n]\}$[18] where, upon initialization each such resource $\mathcal{K}_i$ samples a fresh key and outputs it both to the corresponding sender and to the receiver. Formally, $\mathcal{K}$ is a 2-interface resource which upon initialization samples $k_i \leftarrow_R \mathcal{K}$ and outputs it on both interfaces. The protocol for realizing $\mathcal{F}_{A\text{-}AC}$ from $\mathcal{F}_{A\text{-}IC}$ and $\mathcal{K}_i$ uses a MAC scheme $\mathsf{MAC} = (\mathsf{TAG}, \mathsf{VRFY})$ with message space $\mathcal{M}$. In particular to send a message $m \in \mathcal{M}$ the sender obtains shared key k_i from $\mathcal{K}_i$, computes a tag $\tau = \mathsf{TAG}_{k_i}(m)$ and outputs (m, τ) to $\mathcal{F}_{A\text{-}IC}$ on interface S_i. When the receiver obtains a message of the form (m, τ) from interface R of $\mathcal{F}_{A\text{-}IC}$ it looks for a key k_i (obtained from $\mathcal{K}_i$) such that $\mathsf{VRFY}_{k_i}(m, \tau) = \mathtt{true}$. If such a key is found output (m, i) and otherwise output $\perp$ (on the external interface).

We prove the construction secure using a somewhat involved sequence of hybrid systems as summarized in the following theorem and the proof can be found in Appendix A.2. While the result is not surprising the proof reveals a subtlety arising from the somewhat non-standard use of unforgeability in a multi-user setting. As a consequence, in terms of exact security the construction looses double the expected unforgebaility term ϵ' per sender. The details can be found in Appendix A.2.

Theorem 1. *The trivial protocol* $\pi = (\pi^{S_1}, \cdots, \pi^{S_n}, \rho^R)$ *described above realizes* $\mathcal{F}_{A\text{-}AC}$ *from* $\mathcal{F}_{A\text{-}IC}$ *and* $\mathcal{K}$.

More precisely there exists a simulator σ *such for any* $t, q_t, q'_t, q'_v \in \mathbb{N}$ *and* $\epsilon, \epsilon' > 0$, *distinguisher* D *and MAC scheme* MAC *with message space* $\mathcal{M}$ *such that:*

[18] We use the standard notation $[n]$ to denote the set $\{1, \ldots, n\}$.

– MAC *is* (t, q_t, ϵ)-**ki**-**cma** *secure,* $(t, q'_t, q'_v, \epsilon')$-**uf**-**cmva** *secure and has* η *completeness error.*
– D *runs in time* t*, sends* q'_v *messages through* A*,* $\min(q'_t, \frac{q'_v}{n}, \frac{q_t}{n})$ *messages through* S_i *(for all* $i \in [n]$*).*

we get that $\Delta^{\mathsf{D}}[\pi(\mathcal{F}_{A\text{-}IC}||\mathcal{K}), \sigma^A(\mathcal{F}_{A\text{-}AC})] \leq 2n\epsilon' + q'_v\eta + n\epsilon$.

3.2 Anonymous Secure Channel

An anonymous secure channel (Figure 3) is identical to $\mathcal{F}_{A\text{-}AC}$ except that now only the length $|m|$ of each message sent is leaked on the adversarial interface. In particular while identities of senders are learnt by the receiver they are not leaked to the adversary. Moreover adversaries may only cause delivery of messages previously input by a sender.

INIT: $M \leftarrow \emptyset$, *counter* $\leftarrow 0$
ON INTERFACE S_i:
  CASE (m): *counter* $\leftarrow$ *counter* $+ 1$, $M \leftarrow M \cup \{(\textit{counter}, i, m)\}$; output $(\textit{counter}, |m|)$ on interface A
ON INTERFACE A:
  CASE $(j \in \mathbb{N} \cup \{\perp\})$: If $\exists (j, i, m) \in M$ then output (i, m) on interface R (and otherwise output $\perp$).

Fig. 3. The anonymous secure channel $\mathcal{F}_{A\text{-}SC}$

The main result of this section is to construct $\mathcal{F}_{A\text{-}SC}$ from $\mathcal{F}_{A\text{-}IC}$ and shared keys. As a warm-up we first describe a simple statistically secure construction of $\mathcal{F}_{A\text{-}SC}$ from $\mathcal{F}_{A\text{-}AC}$ which can be combined with the results of the previous section (leveraging the composability of the constructive framework) to achieve our stated goal. However we also provide a direct construction (from $\mathcal{F}_{A\text{-}IC}$ and shared keys) enjoying only computational security but therefore improving on the communication complexity compared to the composed protocol.

Modular Construction. The statistically secure protocol for n senders makes use of $\mathcal{F}_{A\text{-}AC}$ with $2n$ sender interfaces. The i^{th} sender is given access to a (unique) pair of interfaces $S_{i,0}$ and $S_{i,1}$ for $\mathcal{F}_{A\text{-}AC}$. Let λ be the (implicit) security parameter. To send an ℓ-bit message $m = (m_1, \ldots, m_\ell)$ the sender samples a uniform random integer $r \leftarrow_R [2^\lambda]$ and for each $j \in [\ell]$ sends message (r, j) via interfaces S_{i,m_j}. On the receiver's side upon receiving[19], for each $j \in [\ell]$, a message of the form (r_j, j) from identity (i, b_j) if all r_j are equal then output message $m = (b_1, \ldots, b_\ell)$ delivered from sender i. Intuitively the protocol constructs $\mathcal{F}_{A\text{-}SC}$ from $\mathcal{F}_{A\text{-}AC}$ because on the one hand the view of the adversary is independent of all messages m except their length and further, the only chance the adversary has of causing the receiver protocol to output a bad message is if a

[19] delivered in any order

sender reuses the same r for two different messages. Using a standard approximation for the birthday bound the probability that the same value for r is used more than once across q message transmissions can be approximated by $q^2 2^{-\lambda}$.

Direct Construction. Of course when composed with the previous construction of $\mathcal{F}_{A\text{-}AC}$ from $\mathcal{F}_{A\text{-}IC}$ we see that the value of r need only to be sent a single time. Moreover the values of j need not be sent at all. This observation leads to the somewhat more efficient construction of $\mathcal{F}_{A\text{-}SC}$ from $(\mathcal{F}_{A\text{-}IC}||\mathcal{K})$ as follows. Given a $\{\textbf{ki-cma}, \textbf{uf-cmva}\}$-MAC with message space $\mathcal{M} = \mathcal{M}' \times \{0,1\}^{\log \ell}$ each sender is supplied with a pair of keys $k_{i,0}, k_{i,1}$ shared with the receiver. To send message $m \in \{0,1\}^\ell$ the sender computes $c = (r, \{\mathsf{MAC}_{k_{i,m_j}}(r,j)\}_{j\in[\ell]})$ for a randomly sampled $r \leftarrow_R \mathcal{M}'$ which it inputs into interface S_i of $\mathcal{F}_{A\text{-}IC}$. Upon receiving c' the receiver parses it as $c' = (r, \{\tau_j\}_{j\in[\ell]})$ and outputs $(i, m = (m_1, \ldots, m_\ell))$ if and only if it finds keys $k_{i,0}, k_{i,1}$ such that for all $j \in [\ell]$ the tag τ_j is valid for message (r,j) under key k_{i,m_j} where $m_j \in \{0,1\}$.[20]

Theorem 2. *The protocol $\pi = (\pi^{S_1}, \cdots, \pi^{S_n}, \rho^R)$ described above realizes $\mathcal{F}_{A\text{-}SC}$ from $\mathcal{F}_{A\text{-}IC}$ and $\mathcal{K}$.*

More precisely there exists a simulator σ such that for any $t, q_t, q_t', q_v' \in \mathbb{N}$ and $\epsilon, \epsilon' > 0$, distinguisher D and MAC scheme MAC with message space $\mathcal{M}$ such that:

- MAC *is (t, q_t, ϵ)-**ki-cma** secure, $(t, q_t', q_v', \epsilon')$-**uf-cmva** secure and has η completeness error.*
- D *runs in time t, sends q_v messages through A, $\min(q_t', \frac{q_v'}{n}, \frac{q_t}{\ell n})$ messages through S_i (for all $i \in [n]$).*

we get that $\Delta^{\mathsf{D}}[\pi(\mathcal{F}_{A\text{-}IC}||\mathcal{K}), \sigma(\mathcal{F}_{A\text{-}SC})] \leq 4\ell n\epsilon' + \ell q_v' \eta + \frac{q_t'}{|\mathcal{M}'|} + 2n\epsilon' + 2\ell n\epsilon$.

The proof of this theorem, in Appendix A.4, relies on similar hybrid systems as for Theorem 1 followed by a sequence of hybrid systems for formalizing the intuition concerning the statistical security of the construction from $\mathcal{F}_{A\text{-}AC}$. We also point out that the same optimistic optimization as in Appendix A.3 can be applied here to reduce the computational cost of the receiver.

3.3 Anonymous Entity Authentication

We describe a multi-session and multi-user anonymous entity authentication resource $\mathcal{F}_{A\text{-}EA}$ in such a way that we can prove that a standard challenge response protocol indeed constructs it with statistical security.

Resource $\mathcal{F}_{A\text{-}EA}$ models multiple (sequential) authentication sessions initiated via the server interface. Clients respond to the most recent pending authentication challenge whenever prompted to do so via their interface. Each session results either in the server accepting a particular identity or else failing (denoted

[20] If there is more then one value of i this holds then the reciever outputs the smallest such value.

with a special output $\perp$). The adversary, assumed to be controlling the scheduling of the underlying communication channel, is given control over forwarding challenges from the server to the client (via the `QUERY` command). However it learns nothing more than the relative order of responses generated by clients thereby capturing the intuitive goal of anonymity. Further the adversary can, at any point, forward a clients response on to the server.

To capture the intuitive goal of entity authentication we equip $\mathcal{F}_{A\text{-}EA}$ with an internal set *Compromised* which keeps track of the set of clients which have forwarded their response for the *current* authentication session. In particular *Compromised* is cleared whenever a new authentication session is initiated and, crucially, for any given session the adversary can only cause identities contained in *Compromised* to be output on the server's interface. In other words the only identities ever accepted at the end of a session are those which respond *during* the session regardless of all previous actions taken on any interface.[21] A formal description capturing this behavior can be found in Figure 4.

Init: $InSession \leftarrow$ `false`, $counter \leftarrow 0$, $Compromised \leftarrow \emptyset$, $\forall i \in [n]$ $msg_i \leftarrow$ `false`
On Interface C_i:
  Case (`RESPOND`):
    If ($msg_i =$ `true`) then
      $msg_i \leftarrow$ `false`, $counter \leftarrow counter + 1$
      If ($InSession =$ `true`) then $Compromised \leftarrow Compromised \cup \{(counter, i)\}$
      Output $counter$ on interface A
On Interface S:
  Case (`GO`): $InSession \leftarrow$ `true`, $Compromised \leftarrow \emptyset$; output `GO` on interface A
On Interface A:
  Case (`QUERY`): $\forall i \in [n]$ $msg_i \leftarrow$ `true`
  Case ($j \in \mathbb{N} \cup \{\perp\}$):
    If ($InSession =$ `true`) then
      $InSession \leftarrow$ `false`
      If $\exists (j, i) \in Compromised$ then output i on interface S, otherwise output $\perp$ on interface S

Fig. 4. The ideal resource of anonymous entity authentication $\mathcal{F}_{A\text{-}EA}$

To verify that $\mathcal{F}_{A\text{-}EA}$ captures our intended intuition we show that a very simple challenge-response protocol indeed constructs $\mathcal{F}_{A\text{-}EA}$ from $\mathcal{F}_{A\text{-}AC}$ as expected. Subsequently we describe several optimizations of interest for a more practical scenario.

In order to send the challenge from server to clients we assume the presence of a type of single-sender/multi-receiver insecure broadcast channel $\mathcal{F}_{IB}$. Put simply any message input by the sender is output to the adversary and any message input by the adversary is delivered to all receivers.[22] The server protocol

[21] As described in the introduction, in the language of [TM12] this corresponds precisely to $(\{C, S\}, \{S\})$-authenticity.

[22] A formal description can be found in Figure 15 in Appendix A.5.

ρ for construction $\mathcal{F}_{A\text{-}EA}$ is extremely simple. For each new authentication session it chooses a fresh random challenge $r \leftarrow_R \mathcal{M}$ and broadcasts it using $\mathcal{F}_{IB}$. When it receives a response (i, r') from $\mathcal{F}_{A\text{-}AC}$ it outputs identity i if $r' = r$ and otherwise $\perp$. The i^{th} client protocol π^{C_i} is equally simple; it is equipped with a message buffer which stores the most recent message received from $\mathcal{F}_{IB}$. Whenever π receives the command to respond it checks if its message buffer is full and if so forwards the content to interface S_i of $\mathcal{F}_{A\text{-}AC}$. A formal description of this protocol and the proof that it constructs $\mathcal{F}_{A\text{-}EA}$ for $(\mathcal{F}_{A\text{-}AC}||\mathcal{F}_{IB})$ as stated in the following theorem can be found in Appendix A.5.

Theorem 3. *The protocol* $\pi = (\pi^{C_1}, \cdots, \pi^{C_n}, \rho^S)$ *described above realizes* $\mathcal{F}_{A\text{-}EA}$ *from* $\mathcal{F}_{A\text{-}AC}$ *and* $\mathcal{F}_{IB}$.

More precisely, there exists a simulator σ *such for any* $t, q_s, q_v \in \mathbb{N}$*, distinguisher* D *sending* q_v *messages through interface* A *and starting* q_s *sessions, and a challenge set* $\mathcal{M}$ *we get that* $\Delta^{\mathsf{D}}[\pi(\mathcal{F}_{A\text{-}AC}||\mathcal{F}_{IB}), \sigma(\mathcal{F}_{A\text{-}EA})] \leq \frac{q_s(q_s+q_v)}{|\mathcal{M}|}$.

We briefly remark on some variants of this result. Similar to the optimization for building $\mathcal{F}_{A\text{-}SC}$ from $(\mathcal{F}_{A\text{-}IC}||\mathcal{K})$ here too when using $(\mathcal{F}_{A\text{-}IC}||\mathcal{K})$ in place of $\mathcal{F}_{A\text{-}AC}$ the response from the clients need not include the random challenge r. Moreover the same trade-off for the "optimistic setting" described in Appendix A.3 can also be applied here to improve server efficiency using stateful clients. Finally when using KI MACs over $(\mathcal{F}_{A\text{-}IC}||\mathcal{K})$ underneath the challenge-response protocol we observe that it suffices to use only universally unforgeable MACs[23] instead of **uf-cmva** ones. Intuitively, this is because the only messages for which producing a fresh tag could impersonate a client are the random challenges chosen by the server protocol. However for given (t, q_t, q_v, ϵ)-**uf-cmva** secure MAC the exact distinguishing advantage between the real and ideal systems is smaller (by an additive factor of $(n\epsilon - 1)q_s$) then if the MAC is only (t, q_t, q_v, ϵ)-secure against universal forgeries.[24]

This observation can be interpreted in two ways. On the one hand for a given MAC based challenge-response authentication protocol we can weaken the assumptions on the MAC for obtaining secure entity authentication. On the other hand we can make use of potentially more efficient (but slightly more forgeable) MAC schemes for constructing $\mathcal{F}_{A\text{-}EA}$.

References

[ACdM05] G. Ateniese, J. Camenisch, and B. de Medeiros. Untraceable RFID tags via insubvertible encryption. In *ACM CCS*, pages 92–101, 2005.

[23] Universal unforgeability is a relaxed security notion for MACs where the adversary only wins by producing a fresh (valid) tag for a uniform random message chosen by the challenger.

[24] The security loss arises because in addition to having to guess for which client an impersonation attack will arise (see Proposition 1) the reduction to universal unforgeability must also guess during which of the q_s sessions the attack occurs so as to properly plant its random challenge message from the universal unforgeability game.

[AHM+14] J. Alwen, M. Hirt, U. Maurer, A. Patra, and P. Raykov. Key-indistinguishable message authentication codes. Cryptology ePrint Archive, to Appear, 2014.

[AMR+12] M. Arapinis, L. I. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, and R. Borgaonkar. New privacy issues in mobile telephony: fix and verification. In T. Yu, G. Danezis, and V. D. Gligor, editors, *ACM Conference on Computer and Communications Security*, pages 205–216. ACM, 2012.

[AMRR11] M. Arapinis, L. I. Mancini, E. Ritter, and M. Ryan. Formal analysis of umts privacy. *CoRR*, abs/1109.2066, 2011.

[BBDP01] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, pages 566–582, 2001.

[BLdMT09] M. Burmester, T. V. Le, B. de Medeiros, and G. Tsudik. Universally composable RFID identification and authentication protocols. *ACM Trans. Inf. Syst. Secur.*, 12(4), 2009.

[BM11] M. Burmester and J. Munilla. Lightweight RFID authentication with forward and backward security. *ACM Trans. Inf. Syst. Secur.*, 14(1):11, 2011.

[BR93] M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. R. Stinson, editor, *CRYPTO*, pages 232–249, 1993.

[BR05] M. Barbeau and J.-M. Robert. Perfect identity concealment in umts over radio access links. In *WiMob (2)*, pages 72–77. IEEE, 2005.

[Can01] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145, Las Vegas, Nevada, October 2001.

[CRS11] H. Choudhury, B. Roychoudhury, and D. K. Saikia. Umts user identity confidentiality: An end-to-end solution. In *WOCN*, pages 1–6. IEEE, 2011.

[CRS12] H. Choudhury, B. Roychoudhury, and D. K. Saikia. Enhancing user identity privacy in lte. In G. Min, Y. Wu, L. C. Liu, X. Jin, S. A. Jarvis, and A. Y. Al-Dubai, editors, *TrustCom*, pages 949–957. IEEE, 2012.

[DLYZ11] R. H. Deng, Y. Li, M. Yung, and Y. Zhao. A zero-knowledge based framework for RFID privacy. *J. of Computer Security*, 19(6):1109–1146, 2011.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[GQ88] L. C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *EUROCRYPT*, pages 123–128, 1988.

[GVI06] G. Gódor, B. Varadi, and S. Imre. Novel authentication algorithm of future networks. In *ICN/ICONS/MCL*, page 80. IEEE Computer Society, 2006.

[HPVP11] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel. A new RFID privacy model. In *ESORICS*, pages 568–587, 2011.

[KAC08] M. Khan, A. Ahmed, and A. R. Cheema. Vulnerabilities of umts access domain security architecture. In *SNPD*, pages 350–355. IEEE, 2008.

[KMO+13] M. Kohlweiss, U. Maurer, C. Onete, B. Tackmann, and D. Venturi. Anonymity-preserving public-key encryption: A constructive approach. In E. D. Cristofaro and M. Wright, editors, *PETS*, volume 7981 of *LNCS*, pages 19–39. 2013.

[KO05] G. M. Køien and V. A. Oleshchuk. Location privacy for cellular systems; analysis and solution. In G. Danezis and D. Martin, editors, *Privacy Enhancing Technologies*, volume 3856 of *LNCS*, pages 40–58. Springer, 2005.

[LSWW13] M.-F. Lee, N. P. Smart, B. Warinschi, and G. Watson. Anonymity guarantees of the umts/lte authentication and connection protocol. Cryptology ePrint Archive, Report 2013/027, 2013. `http://eprint.iacr.org/`.

[Mau02] U. Maurer. Indistinguishability of random systems. In L. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. May 2002.

[Mau11] U. Maurer. Constructive cryptography - a new paradigm for security definitions and proofs. In S. Mödersheim and C. Palamidessi, editors, *TOSCA*, volume 6993 of *LNCS*, pages 33–56. Springer, 2011.

[MT12] P. Mol and S. Tessaro. Manuscript, Dec. 2012.

[Oka92] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, pages 31–53, 1992.

[rGPP12] 3rd Generation Partnership Project. Ts 33.102 - 3g security; Security architecture v11.5.0, 2012.

[SAJ07] B. Sattarzadeh, M. Asadpour, and R. Jalili. Improved user identity confidentiality for umts mobile networks. In *ECUMN*, pages 401–409. IEEE, 2007.

[Sch89] C.-P. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, pages 239–252, 1989.

[TM12] J.-K. Tsay and S. F. Mjølsnes. A vulnerability in the umts and lte authentication and key agreement protocols. In I. V. Kotenko and V. A. Skormin, editors, *MMM-ACNS*, volume 7531 of *LNCS*, pages 65–76. Springer, 2012.

[Vau10] S. Vaudenay. Privacy models for RFID schemes. In *RFIDSec*, page 65, 2010.

A Real/Ideal Transformation

This section contains the missing proofs and details for Section 3.

A.1 Technical Lemmata

We prove the following lemma which formalizes the exact security obtained from a standard hybrid argument.

Lemma 1. *For any three systems $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ with $\mathcal{S}_1 \approx_\alpha \mathcal{S}_2 \approx_\beta \mathcal{S}_3$ (where $\alpha = (t, q_t, q_v, \epsilon)$ and $\beta = (t', q'_t, q'_v, \epsilon')$) it holds that $\mathcal{S}_1 \approx_\gamma \mathcal{S}_3$ where $\gamma = (\min(t, t'), \min(q_t, q'_t), \min(q_v, q'_v), \epsilon + \epsilon')$.*

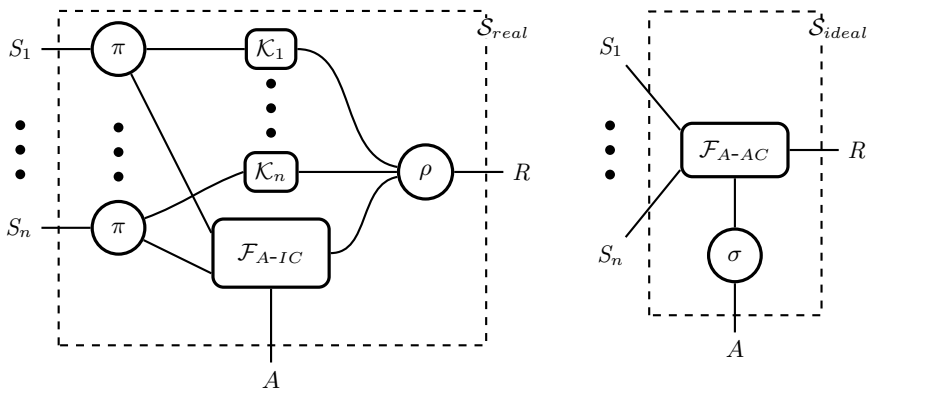
Proof. Take any $(\min(t, t'), \min(q_t, q'_t), \min(q_v, q'_v))$-distinguisher D. The triangle equality gives us that

$$\Delta^{\mathsf{D}}(\mathcal{S}_1, \mathcal{S}_3) \leq \Delta^{\mathsf{D}}(\mathcal{S}_1, \mathcal{S}_2) + \Delta^{\mathsf{D}}(\mathcal{S}_2, \mathcal{S}_3).$$

It remains to notice that $\Delta^{\mathsf{D}}(\mathcal{S}_1, \mathcal{S}_2)$ is bounded by ϵ since D is a (t, q_t, q_v)-distinguisher. Similarly $\Delta^{\mathsf{D}}(\mathcal{S}_2, \mathcal{S}_3)$ is bounded by ϵ' since D is a (t', q'_t, q'_v)-distinguisher. □

Next we introduce a technical lemma from [Mau02] used to lower bound the advantage of a distinguisher when two systems behave identically unless a particular "bad" event occurs.

Lemma 2. *Let E be an event defined over the states of system $\mathcal{S}_1$. Further suppose systems $\mathcal{S}_1$ and $\mathcal{S}_2$ behave identically if E is not triggered (by D) for $\mathcal{S}_1$. Then for any distinguisher D we have that $\Delta^{\mathsf{D}}(\mathcal{S}_1, \mathcal{S}_2) \leq \Pr[E]$.*



(a) Real world (b) Ideal world with the simulator σ

Fig. 5. Systems for describing message authentication

A.2 Proof of Theorem 1

Proof. The claim made in the theorem is depicted in Figure 5.

We use a shortcut notation $\alpha = (t, q_t, q_v, \epsilon)$ instead of $(t, \underbrace{q_t, \ldots, q_t}_{n \text{ times}}, q_v, \epsilon)$ when talking about distinguishing $\pi(\mathcal{F}_{A\text{-}IC}||\mathcal{K})$ and $\sigma^A\mathcal{F}_{A\text{-}AC}$. Formally, $\pi(\mathcal{F}_{A\text{-}IC}||\mathcal{K}) \approx_\alpha \sigma^A\mathcal{F}_{A\text{-}AC}$ means that any D making at most q_t queries to each interface S_i and q_v queries to the interface A has advantage at most ϵ.

We define the simulator σ followed by systems $\mathcal{S}_{real}, \mathcal{S}_{ideal}, \mathcal{H}_1$ and $\mathcal{H}_2$ such that we can show that:

$$\pi(\mathcal{F}_{A\text{-}IC}||\mathcal{K}) = \mathcal{S}_{real} \overset{1}{\approx}_\alpha \mathcal{H}_1 \overset{2}{\approx}_\beta \mathcal{H}_2 \overset{3}{\approx}_\gamma \mathcal{S}_{ideal} = \sigma^A\mathcal{F}_{A\text{-}AC}.$$

where $\alpha = (t, q'_t, q'_v, n\epsilon' + q'_v\eta)$, $\beta = (t, \min(q'_t, \frac{q'_v}{n}), q'_v, n\epsilon')$, $\gamma = (t, \frac{q_t}{n}, q'_v, n\epsilon)$.

The first and last equalities follow by inspection of the definitions of all involved resources, protocols and the simulator as they are merely the result of rewriting the logic of those systems. Once the descriptions are complete it remains to prove the intermediate equalities. In particular equalities hold because:

(1) follows from the **uf-cmva** property of MAC (Proposition 1).
(2) follows from the **uf-cmva** property of MAC (Proposition 2).
(3) follows from the **ki-cma** property (Proposition 3).

The theorem now follows via a standard hybrid argument as stated in Lemma 1. □

The Simulator

The simulator, described in Figure 6, has 2 interfaces: the inner interface F for interacting with $\mathcal{F}_{A\text{-}AC}$, and the outer interface A to interact with the adversary. Essentially the simulator fully emulates all clients internally as well as the server as they behave in the real world. Crucial though, the simulator does not learn the identity of any responding clients and so it can not produce tags appropriately. Instead the simulator simply uses a single MAC key to produce all responses.

We use the KI property of the MAC to show that this is still a valid simulation strategy.

The other crucial difference is that in the ideal world the resource $\mathcal{F}_{A\text{-}AC}$, by definition, enforces bounds on which messages can be authenticated (i.e. only those which were sent). On the other hand, in the real world where an adversary can cause *any* message to be authenticated as long as it can supply a corresponding valid tag. We use the unforgeability property of the MAC to show that the two worlds are essentially the same.

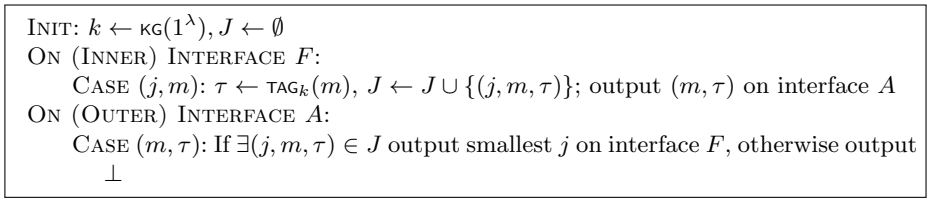


Fig. 6. The simulator σ

Defining $\mathcal{S}_{real}$ and $\mathcal{S}_{ideal}$

In this section we describe how real and ideal systems behave when seen as single monolithic systems with $n + 2$ interfaces $S_1, \ldots, S_n$, A and R i.e. from the point of view of the distinguisher. In other words in Figure 7 we describe the system $\mathcal{S}_{real}$ which is obtained by composing all protocols and resources of $\pi^{S_1} \cdots \pi^{S_n} \rho^R(\mathcal{F}_{A\text{-}IC}||\mathcal{K}_1|| \cdots ||\mathcal{K}_n)$ (and simplifying some redundant logic as described below). Similarly in Figure 8 we describe $\mathcal{S}_{ideal}$ which captures the behavior of $\sigma^A(\mathcal{F}_{A\text{-}AC})$.

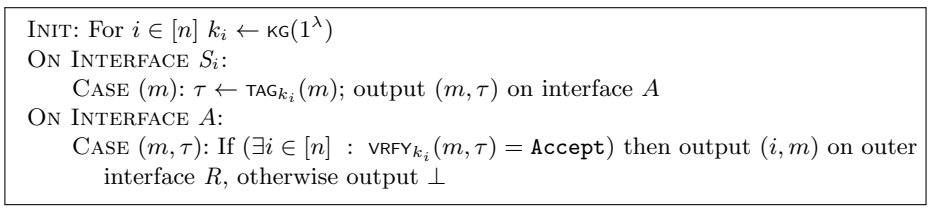


Fig. 7. The essential functionality $\mathcal{S}_{real}$ of $\mathcal{F}_{A\text{-}IC}$ with protocols for senders and the receiver

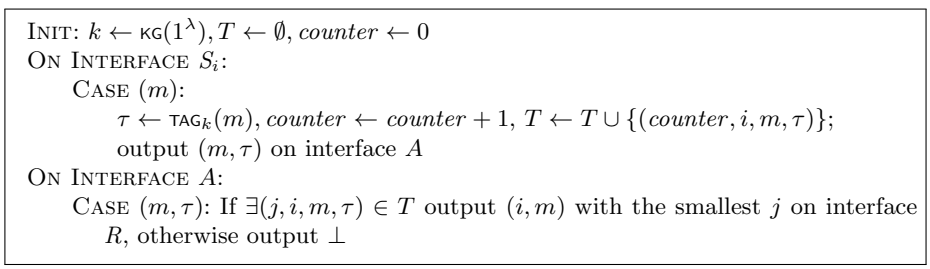


Fig. 8. The essential functionality $\mathcal{S}_{ideal}$ of $\mathcal{F}_{A\text{-}AC}$ with σ

The fact for any distinguisher D we have

$$\Delta^{\mathsf{D}}\left(\pi^{S_1} \cdots \pi^{S_n} \rho^R(\mathcal{F}_{A\text{-}IC}||\mathcal{K}_1|| \cdots ||\mathcal{K}_n), \mathcal{S}_{real}\right) = 0 \text{ and } \Delta^{D}\left(\sigma^A \mathcal{F}_{A\text{-}AC}, \mathcal{S}_{ideal}\right) = 0$$

follows by inspection of the definitions of resources as in both cases the systems exhibit identical I/O behavior.

Defining $\mathcal{H}_1$ and Proving $\mathcal{S}_{real} \approx \mathcal{H}_1$

We define system $\mathcal{H}_1$ (see Figure 9 for a detailed description) to be identical to $\mathcal{S}_{real}$ with the difference that it changes the verification mechanism of messages.

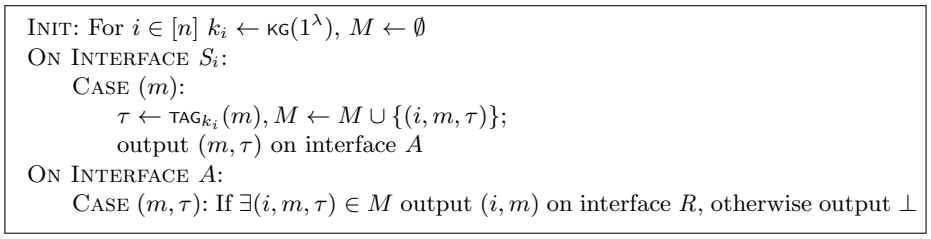


Fig. 9. Hybrid functionality $\mathcal{H}_1$

More precisely, it has the following modifications:

1. It has a special variable M which keeps track of all messages that have been authenticated. The variable M is initialized with $\emptyset$ in the INIT stage, and gets a triple (i, m, τ) appended whenever the client i authenticates a message m with a tag τ.
2. When processing a message (m, τ) input on interface A, the system $\mathcal{H}_1$ does not use verification algorithm VRFY of MAC scheme anymore. Instead it checks if for some client i a triple (i, m, τ) belongs to M. In the positive case it concludes that for this triple $\mathsf{VRFY}_{k_i}(m, \tau) = \texttt{Accept}$ anyway and outputs i on interface R. In the negative case only $\perp$ is output on interface R.

In this subsection we prove the following proposition.

Proposition 1. *For $t, q_t, q_v \in \mathbb{N}$ and $\epsilon \geq 0$, if* MAC *is (t, q_t, q_v, ϵ)-***uf-cmva** *secure and has completeness error η then $\mathcal{S}_{real} \approx_\alpha \mathcal{H}_1$ where $\alpha = (t, q_t, q_v, n\epsilon + q_v\eta)$.*

Proof. On the highest level the proof has the following structure. We introduce events F (for "forgery") and E (for "completeness error") that may occur when a distinguisher interacts with $\mathcal{H}_1$. We show that if events F and E do not occur systems $\mathcal{H}_1$ and $\mathcal{S}_{real}$ exhibit the same input/output behavior. Then, we give upper bounds for probabilities $\Pr[F]$ and $\Pr[E]$. The first probability is upper bounded using the fact that MAC is **uf-cmva** secure and the latter is upper bounded based on the fact that MAC has sufficiently small completeness error η. Finally, we apply Lemma 2 to upper bound distinguishing advantage for $\mathcal{H}_1$ and $\mathcal{S}_{real}$.

We define F to happen when adversary inputs a message (m, τ) on interface A such that:

$$\exists i \in [n] \ : \ \mathsf{VRFY}_{k_i}(m, \tau) = \texttt{Accept} \quad \text{and} \quad (i, m, \tau) \notin M.$$

We define E to happen when adversary inputs a message (m, τ) on interface A such that:

$$\exists i \in [n] \ : \ \mathsf{VRFY}_{k_i}(m, \tau) = \texttt{Reject} \quad \text{and} \quad (i, m, \tau) \in M.$$

Claim 1. *For any* D *interacting with $\mathcal{H}_1$ and $\mathcal{S}_{real}$ if the event $(F \vee E)$ does not happen in $\mathcal{H}_1$ then both systems behave the same for* D.

Proof. If F and E do not occur, then we know that when $\mathcal{H}_1$ executes the code

If $\exists i \in [n] \ : \ (i, m, \tau) \in M$ then output (i, m) on interface R

the following two facts hold:

$(\neg F) \quad \forall i \in [n] \ : \ \mathsf{VRFY}_{k_i}(m, \tau) = \mathtt{Accept} \Longrightarrow (i, m, \tau) \in M$

$(\neg E) \quad \forall i \in [n] \ : \ (i, m, \tau) \in M \Longrightarrow \mathsf{VRFY}_{k_i}(m, \tau) = \mathtt{Accept}$

We see that two conditions $(i, m, \tau) \in M$ and $\mathsf{VRFY}_{k_i}(m, \tau) = \mathtt{Accept}$ are equivalent in this case and hence we can substitute one with another in the definition of $\mathcal{H}_1$. If we make such a substitution we get exactly the same system as $\mathcal{S}_{real}$. Finally, we have that $\mathcal{H}_1$ behaves the same as[25] $\mathcal{S}_{real}$ as long as F and E do not happen. □

Claim 2. $\Pr[F] \leq n\epsilon$.

Proof. We give a reduction $\mathsf{R}(\mathsf{D})$ which wins **uf-cmva** game for MAC with probability $\frac{\Pr[F]}{n}$. The reduction R works as follows:

1. Given two oracles $\mathsf{TAG}_k(\cdot)$ and $\mathsf{VRFY}_k(\cdot, \cdot)$ reduction R samples uniform random $j \leftarrow [n]$.
2. Reduction R instantiates system $\mathcal{H}_1$ with the the following two differences: (a) All tag queries for k_j made by $\mathcal{H}_1$ are answered with calls to $\mathsf{TAG}_k(\cdot)$; (b) Whenever $\mathcal{H}_1$ checks that $(j, m, \tau) \in M$ it queries $\mathsf{VRFY}_k(\cdot, \cdot)$ oracle with a pair (m, τ).
3. Reduction R lets D interact with $\mathcal{H}_1$.

The reduction R wins $\mathbf{Exp}^{\textbf{uf-cmva}}_{\mathsf{MAC}}(\mathsf{R}(\mathsf{D}), \lambda)$ game in case F is triggered for $i = j$. Assuming F happens, the probability that this condition is met is exactly $1/n$ (as j is chosen uniformly and independently of D's view). Hence, the probability that R outputs a valid forgery is at least $\frac{\Pr[F]}{n}$. Finally, if D is (t', q'_t, q'_v)-distinguisher then $\mathsf{R}(\mathsf{D})$ is a (t', q'_t, q'_v)-adversary. Since MAC is (t, q_t, q_v, ϵ)-**uf-cmva** secure then $\frac{\Pr[F]}{n} \leq \epsilon$ as is desired. □

Claim 3. $\Pr[E] \leq q_v\eta$.

Proof. For every individual invocation of the VRFY procedure, the probability that a tag which was produced by the TAG algorithm is not verified is upper bounded by η. In total VRFY algorithm is invoked at most q_v times, so the probability of at least one completeness error happening is upper bounded by $q_v\eta$. □

Now we use Lemma 2 to upper bound distinguishing advantage of D as $\Delta^{\mathsf{D}}(\mathcal{H}_1, \mathcal{S}_{real}) \leq \Pr[F \vee E]$ which is smaller or equal to $n\epsilon + q_v\eta$. □

Defining $\mathcal{H}_2$ and Proving $\mathcal{H}_1 \approx \mathcal{H}_2$

We define system $\mathcal{H}_2$ to be identical to $\mathcal{S}_{ideal}$ with the difference that it has n keys instead of 1. We notice that the system $\mathcal{H}_2$ is different from $\mathcal{H}_1$ in employing *counter* to count the authenticated messages.

Proposition 2. *For any $t, q_t, q_v \in \mathbb{N}$ and $\epsilon \geq 0$, if* MAC *is $(t, q_t, q_v\epsilon)$-***uf-cmva** *secure then $\mathcal{H}_1 \approx_\alpha \mathcal{H}_2$ where $\alpha = (t, \min(q_t, q_v/n), q_v, n\epsilon)$.*

[25] Formally: exhibits the same I/O distribution as...

INIT: For $i \in [n]$ $k_i \leftarrow \mathsf{KG}(1^\lambda)$, $T \leftarrow \emptyset$
ON INTERFACE S_i:
  CASE (m):
    $\tau \leftarrow \mathsf{TAG}_{k_i}(m)$, $counter \leftarrow counter + 1$, $T \leftarrow T \cup \{(counter, i, m, \tau)\}$;
    output (m, τ) on interface A
ON INTERFACE A:
  CASE (m, τ): If $\exists (j, i, m, \tau) \in T$ output (i, m) with the smallest j on interface R, otherwise output $\perp$

Fig. 10. Hybrid functionality $\mathcal{H}_2$

Proof. On the highest level the proof has the following structure. We introduce event F (for "cross-forgery") that may occur when a distinguisher interacts with $\mathcal{H}_2$. We show that if event F does not occur systems $\mathcal{H}_1$ and $\mathcal{H}_2$ exhibit the same input/output behavior. Then, we give upper bounds for probability $\Pr[F]$ using the fact that MAC is **uf-cmva** secure. Finally, we apply Lemma 2 to upper bound distinguishing advantage for $\mathcal{H}_1$ and $\mathcal{H}_2$.

We define F to happen when a new entry (j, i, m, τ) is added to T such that an entry (j', i', m, τ) is already in T for some $i' \neq i$.

Claim 4. *For any* D *interacting with* $\mathcal{H}_1$ *and* $\mathcal{H}_2$ *if the event* F *does not happen in* $\mathcal{H}_2$ *then both systems behave the same for* D.

Proof. In this case we have that the variable *counter* is irrelevant for the $\mathcal{H}_2$'s behavior. After removing it, we get exactly $\mathcal{H}_1$. □

Claim 5. $\Pr[F] \leq n\epsilon$.

Proof. The proof is similar to one in Claim 2. We give a reduction $\mathsf{R}(\mathsf{D})$ which wins **uf-cmva** game for MAC with probability $\frac{\Pr[F]}{n}$. The reduction R works as follows:

1. Given two oracles $\mathsf{TAG}_k(\cdot)$ and $\mathsf{VRFY}_k(\cdot, \cdot)$ reduction R samples uniform random $j \leftarrow [n]$.
2. Reduction R instantiates system $\mathcal{H}_2$ with the the following two differences: (a) All tag queries for k_j made by $\mathcal{H}_2$ are answered with calls to $\mathsf{TAG}_k(\cdot)$; (b) Whenever $\mathcal{H}_2$ produces a new tag τ for a message m for identity different from j it queries $\mathsf{VRFY}_k(\cdot, \cdot)$ oracle with a pair (m, τ).
3. Reduction R lets D interact with $\mathcal{H}_2$.

The reduction R wins $\mathbf{Exp}^{\textbf{uf-cmva}}_{\mathsf{MAC}}(\mathsf{R}(\mathsf{D}), \lambda)$ game in case cross-forgery event F is triggered for the j^{th} sender. Assuming F happens, the probability that this condition is met is exactly $1/n$ (as j is chosen uniformly and independently of D's view). Hence, the probability that R produces a valid cross-forgery is at least $\frac{\Pr[F]}{n}$. Finally, if D is (t', q'_t, q'_v)-distinguisher then $\mathsf{R}(\mathsf{D})$ is a (t', q'_t, nq'_t)-adversary. Since MAC is (t, q_t, q_v, ϵ)-**uf-cmva** secure then $\frac{\Pr[F]}{n} \leq \epsilon$ as is desired. □

Finally, we conclude that $\alpha = (t, \min(q_t, q_v/n), q_v, n\epsilon)$. □

Prove that $\mathcal{H}_2 \approx \mathcal{S}_{ideal}$

In this section we prove that the system $\mathcal{H}_2$ is indistinguishable from $\mathcal{S}_{ideal}$. The only difference between two systems is that $\mathcal{H}_2$ uses n keys to tag messages instead of one as $\mathcal{S}_{ideal}$.

Proposition 3. *For any $t, q_t, q_v \in \mathbb{N}$ and $\epsilon \geq 0$, if* MAC *is (t, q_t, ϵ)-***ki-cma** *secure then $\mathcal{H}_2 \approx_\alpha \mathcal{S}_{ideal}$ where $\alpha = \left(t, \frac{q_t}{n}, q_v, n\epsilon\right)$.*

Proof. We prove this proposition by a standard hybrid argument with $n-2$ intermediate systems. In order to describe intermediate systems we use the following notation. By $L := [k_1, \ldots, k_n]$ we denote a list of TAG oracles corresponding to the keys $k_1, \ldots, k_n$. For some list L of n TAG oracles we define $\mathcal{Q}(L)$ to be a system identical to $\mathcal{H}_2$ with the following difference: for all i, it substitutes calls to oracle TAG_{k_i} with calls to the i^{th} oracle in the list L. For each $i \in [n]$, we define the following "intermediate" lists of oracles $L_i = [\underbrace{k, \ldots, k}_{i \text{ times}}, k_{i+1}, k_{i+2}, \ldots, k_n]$. In particular we have that $\mathcal{Q}(L_1) = \mathcal{H}_2$ and $\mathcal{Q}(L_n) = \mathcal{S}_{ideal}$.

Lemma 3. *For any $t, q_t, q_v \in \mathbb{N}$, $\epsilon \geq 0$ and $i \in [n-1]$, if* MAC *is (t, q_t, ϵ)-***ki-cma** *secure then $\mathcal{Q}(L_i) \approx_{\alpha_i} \mathcal{Q}(L_{i+1})$ where $\alpha_i = \left(t, \frac{q_t}{i}, q_v, \epsilon\right)$.*

Proof. We prove Lemma by presenting a reduction R which, given a distinguisher D for $\mathcal{Q}(L_i)$ and $\mathcal{Q}(L_{i+1})$ with some advantage, wins the **ki-cma** game with at least the same advantage. The reduction $\mathsf{R}(\mathsf{D})$ works as follows:

1. R is given two oracles $[k_0]$ and $[k_b]$.
2. For each $j \in [i+2, n]$ reduction R generates $k_j \leftarrow \mathsf{KG}(1^\lambda)$.
3. The reduction R creates the list of oracles $L := [\underbrace{k_0, \ldots, k_0}_{i \text{ times}}, k_b, k_{i+2}, \ldots, k_n]$.
4. The reduction R instantiates system $\mathcal{Q}(L)$ and lets D interact with it. Whenever D outputs a bit b', R outputs b' as well.

It remains to notice that if $b = 0$ then R perfectly simulates D interacting with $\mathcal{Q}(L_i)$ and when $b = 1$ then R perfectly simulates D interacting with $\mathcal{Q}(L_{i+1})$. Finally, if D is (t', q'_t, q'_v, q'_s)-distinguisher then $\mathsf{R}(\mathsf{D})$ is a (t', iq'_t)-adversary. The factor (i) appears before q'_t because the TAG oracle of k_0 can be called q'_t times when it is addressed as any of the first i oracles of L_i or L_{i+1}. □

Due to the Lemma above we get the following sequence of indistinguishability statements

$$\mathcal{H}_2 = \mathcal{Q}(L_1) \approx_{\alpha_1} \mathcal{Q}(L_2) \approx_{\alpha_2} \cdots \approx_{\alpha_{n-1}} \mathcal{Q}(L_n) = \mathcal{S}_{ideal}.$$

It remains to notice that according to Lemma 1 this implies $\mathcal{H}_2 \approx_\alpha \mathcal{S}_{ideal}$ where $\alpha = \left(t, \frac{q_t}{n}, q_v, n\epsilon\right)$.

□

A.3 Optimistic Verification

We describe a trade-off which potentially allows us to greatly improve the receivers efficiency without impacting the security of the protocol but at the added cost of a stateful sender in certain real world applications. In particular by maintaining synchronized state between sender and receiver we reduce the cost of receiving a message in the (optimistic but often realistic) case when the adversary did not tamper with messages en-route from n MAC verifications to 1 verification and a PRG evaluation since the last synchronization.

Very briefly, a seed $s_i \in \mathcal{S}$ for PRG $g : \mathcal{S} \to \mathcal{S} \times \mathcal{R}$ is shared between each sender and the receiver. For each message sent the PRG is evaluated $g(s_i) =$

(s'_i, r), the seed updated $s_i \leftarrow s'_i$ and the value (m, τ, r) is sent. The receiver maintains a table with entries (i, k_i, s_i, r). Upon receiving $(m, \tau, \bar{r})$ it checks for an entry with matching $\bar{r}$ and, if found, whether $\mathsf{VRFY}_{k_i}(m, \tau) = \texttt{true}$. If so it evaluates $g(s_i) = (s'_i, r')$, updates the entry to (i, k_i, s'_i, r') and outputs (i, m). Anonymity is preserved since the output of a PRG is pseudo-random and thus indistinguishable between seeds and no value of r is reused for the same seed.

From a theoretical perspective no security is lost. But from a practical perspective the solution helps in a setting where messages are usually delivered undisturbed and/or where re-synchronization between sender and receiver can be performed in a non-anonymous environment. A similar idea was introduced in the context of RFID authentication in [BLdMT09] but using a different technique. In comparison, our main drawback is that we require a stateful sender[26] and the adversary is now able to desynchronize sender and receiver to the (limited) effect of forcing them to revert to the old verification procedure (for that sender only) until a resynchronization procedure is performed. On the other hand our approach requires less computation on the sender side and more significantly does not require the receiver to (pre)compute n PRF values so as to update its database after each successful authentication session.

A.4 Proof of Theorem 2

Proof. We use the same shortcut notation $\alpha = (t, q_t, q_v, \epsilon)$ in the context of distinguishing $\pi(\mathcal{F}_{A\text{-}IC}||\mathcal{K})$ and $\sigma^A \mathcal{F}_{A\text{-}SC}$ as in the proof of Theorem 1 in Appendix A.2.

We define the simulator σ followed by two systems $\mathcal{S}_{real}, \mathcal{S}_{ideal}$ and a hybrid $\mathcal{H}$ such that we can show that:

$$\pi(\mathcal{F}_{A\text{-}IC}||\mathcal{K}) = \mathcal{S}_{real} \overset{1}{\approx}_{\alpha} \mathcal{H}_1 \overset{2}{\approx}_{\beta} \mathcal{H}_2 \overset{3}{\approx}_{\gamma} \mathcal{S}_{ideal} = \sigma^A \mathcal{F}_{A\text{-}SC}.$$

where $\alpha = (t, \min(q'_t, q'_v), q'_v, 4\ell n\epsilon' + \ell q'_v \eta + \frac{q'_t}{|\mathcal{M}'|})$, $\beta = (t, \min(q'_t, \frac{q'_v}{n}), q'_v, 2n\epsilon')$, $\gamma = (t, \frac{q_t}{\ell n}, q'_v, 2\ell n\epsilon)$.

The first and last equalities follow by inspection of the definitions of all involved resources, protocols and the simulator as they are merely the result of rewriting the logic of those systems. Once the descriptions are complete it remains to prove the intermediate equalities. In particular equalities hold because:

(1) follows from the **uf-cmva** property of MAC (Proposition 4).
(2) follows from the **uf-cmva** property of MAC (Proposition 5).
(3) follows from the **ki-cma** property (Proposition 6).

The theorem now follows via a standard hybrid argument as stated in Lemma 1. □

The Simulator

We define the simulator σ in Figure 11, similar to the one in Section A.2.

[26] We note that for lightweight devices such as RFID and USIM cards in mobile phones maintaining state may be considered less expensive then sampling random values in which case this is not an added requirement since presumably a stateful PRG will already be used to implement the randomness required for the tagging operation.

INIT: $k \leftarrow \mathsf{KG}(1^\lambda), J \leftarrow \emptyset$
ON (INNER) INTERFACE F:
CASE $(j, |m|)$: $r \leftarrow_R \mathcal{M}'$, $e \leftarrow (r, \{\mathsf{TAG}_k(r,i)\}_{i\in[|m|]})$, $J \leftarrow J \cup \{(j,e)\}$; output (j,e) on interface A
ON (OUTER) INTERFACE A:
CASE (e): If $\exists (j,e) \in J$ output smallest j on interface $\mathcal{F}_{A\text{-}AC}$, otherwise output $\perp$

Fig. 11. The simulator σ

Defining $\mathcal{S}_{real}$ and $\mathcal{S}_{ideal}$

In this section we describe how systems real and ideal systems behave when seen as single monolithic systems with $n+2$ interfaces $S_1, \ldots, S_n$, A and R i.e. from the point of view of the distinguisher. In other words in Figure 12 we describe the system $\mathcal{S}_{real}$ which is obtained by composing all protocols and resources of $\pi^{S_1} \cdots \pi^{S_n} \rho^R(\mathcal{F}_{A\text{-}IC}||\mathcal{K}_1|| \cdots ||\mathcal{K}_n)$ (and simplifying some redundant logic as described below). Similarly in Figure 13 we describe $\mathcal{S}_{ideal}$ which captures the behavior of $\sigma^A(\mathcal{F}_{A\text{-}SC})$.

INIT: For $i \in [n]$, $k_0^i, k_1^i \leftarrow \mathsf{KG}(1^\lambda)$
ON INTERFACE S_i:
CASE (m): $r \leftarrow_R \mathcal{M}'$, $e \leftarrow (r, \{\mathsf{TAG}_{k^i_{m_j}}(r,j)\}_{j\in[|m|]})$; output e on interface A
ON INTERFACE A:
CASE $(r, \tau, \ldots, \tau_\ell)$:
If $(\exists i \in [n], m \in \{0,1\}^\ell \ : \ \forall j \in [\ell]\ \mathsf{VRFY}_{k^i_{m_j}}(r, \tau_j) = \mathtt{Accept})$ then output (i,m) (with the smallest such i) on outer interface R
Otherwise output $\perp$ on outer interface R

Fig. 12. The essential functionality $\mathcal{S}_{real}$ of $\mathcal{F}_{A\text{-}IC}$ with protocols for senders and the receiver

INIT: $M \leftarrow \emptyset$, $counter \leftarrow 0$, $k \leftarrow \mathsf{KG}(1^\lambda)$
ON INTERFACE S_i:
CASE (m): $counter \leftarrow counter + 1$, $r \leftarrow_R \mathcal{M}'$, $e \leftarrow (r, \{\mathsf{TAG}_{k^i_{m_j}}(r,j)\}_{j\in[|m|]})$, $M \leftarrow M \cup \{(counter, i, m, e)\}$; output e on interface A
ON INTERFACE A:
CASE (e):
If $\exists (j,i,m,e) \in J$ output (i,m) with the smallest j on interface R, otherwise output $\perp$

Fig. 13. The essential functionality $\mathcal{S}_{ideal}$ of $\mathcal{F}_{A\text{-}SC}$ with σ

Defining $\mathcal{H}_1, \mathcal{H}_2$ and Proving $\mathcal{S}_{real} \approx \mathcal{H}_1 \approx \mathcal{H}_2 \approx \mathcal{S}_{ideal}$

We define system $\mathcal{H}_1$ (see Figure 14 for a detailed description) to be identical to $\mathcal{S}_{real}$ with the difference that it changes the verification mechanism of messages.

More precisely, it has the following modifications:

1. It has a special variable M which keeps track of all ciphertexts that have been sent. The variable M is initialized with $\emptyset$ in the INIT stage, and gets a triple (i,m,e) appended whenever the client i encrypts m as e.

INIT: For $i \in [n]$, $k_0^i, k_1^i \leftarrow \mathsf{KG}(1^\lambda)$, $M \leftarrow \emptyset$
ON INTERFACE S_i:
CASE (m):
CASE (m): $r \leftarrow_R \mathcal{M}'$, $e \leftarrow (r, \{\mathsf{TAG}_{k^i_{m_j}}(r, j)\}_{j \in [|m|]})$,
$M \leftarrow M \cup \{(i, m, e)\}$; output e on interface A
ON INTERFACE A:
CASE (e): If $\exists (i, m, e) \in M$ output (i, m) on interface R, otherwise output $\perp$

Fig. 14. Hybrid functionality $\mathcal{H}_1$

2. When processing a ciphertext e input on interface A, the system $\mathcal{H}_1$ does not use verification algorithm VRFY of MAC scheme anymore. Instead it checks if for some client i and a message m a triple (i, m, e) belongs to M. In the positive case it concludes e is an encoding of m by the sender S_i and outputs (i, m) on interface R. In the negative case only $\perp$ is output on interface R.

Proposition 4. *For $t, q_t, q_v \in \mathbb{N}$ and $\epsilon \geq 0$, if* MAC *is (t, q_t, q_v, ϵ)-**uf**-**cmva** secure and has completeness error η then $\mathcal{S}_{real} \approx_\alpha \mathcal{H}_1$ where $\alpha = \left(t, \min(q_t, q_v), q_v, 4\ell n\epsilon + \ell q_v \eta + \frac{q_t}{|\mathcal{M}'|}\right)$.*

Proof. The proof is similar to the one for Proposition 1. Next we highlight the main differences:

1. The probability of not verifying correct message is $\ell q_v \eta$.
2. The probability of forging a message which was not sent is $2\ell n\epsilon$.
3. There is now probability of having two messages with the same randomness which is $\frac{q_t}{|\mathcal{M}'|}$.
4. The probability that an entry τ_i in ciphertext e is good for both k_0, k_1 is at most $2\ell n\epsilon$ for (t, q_t, q_t)-adversary.

□

The system $\mathcal{H}_2$ is defined in the similar spirit as before by introducing a counter j for each message sent secretly.

Proposition 5. *For any $t, q_t, q_v \in \mathbb{N}$ and $\epsilon \geq 0$, if* MAC *is $(t, q_t, q_v\epsilon)$-**uf**-**cmva** secure then $\mathcal{H}_1 \approx_\alpha \mathcal{H}_2$ where $\alpha = (t, \min(q_t, q_v/n), q_v, 2n\epsilon)$.*

Proof. As before we introduce cross-forgery event for a ciphertext e in case it can be ciphertext for two different senders. The analysis is the same, we put the oracle in place of the random tag used to encode the first of the message. □

Proposition 6. *For any $t, q_t, q_v \in \mathbb{N}$ and $\epsilon \geq 0$, if* MAC *is (t, q_t, ϵ)-**ki**-**cma** secure then $\mathcal{H}_2 \approx_\alpha \mathcal{S}_{ideal}$ where $\alpha = \left(t, \frac{q_t}{\ell n}, q_v, 2\ell n\epsilon\right)$.*

Proof. We do a longer hybrid argument now, consisting of $2\ell n$ times substitution, similar to Proposition 3. □

A.5 Proof of Theorem 3

The formal description of our insecure broadcast resource and the client and server protocols can be found in Figures 15, 16 and 17 respectively.

The remainder of this section is dedicated to proving Theorem 3.

On Interface S:
  Case (m): Output m on interface A
On Interface A:
  Case (m): Output m on interfaces $R_1, \ldots, R_n$

Fig. 15. The insecure broadcast channel $\mathcal{F}_{IB}$

Init: *buffer* $\leftarrow \perp$
On Interface R_i of $\mathcal{F}_{IB}$:
  Case (m): *buffer* $\leftarrow m$
On Outer Interface:
  Case (`RESPOND`): If (*buffer* $\neq \perp$) then set *buffer* $\leftarrow \perp$ and output *buffer* on interface S_i of $\mathcal{F}_{A\text{-}AC}$

Fig. 16. The i^{th} client protocol π

Init: $rc \leftarrow \perp$
On Outer Interface:
  Case (`GO`): Output $rc \leftarrow_R \mathcal{M}$ on interface S to $\mathcal{F}_{IB}$
On Interface R to $\mathcal{F}_{A\text{-}AC}$:
  Case (i, m): If $(rc \neq \perp)$ and $(rc = m)$ then set $rc \leftarrow \perp$ and output i on the outer interface

Fig. 17. The server protocol ρ

Proof. We define the simulator σ followed by two systems $\mathcal{S}_{real}, \mathcal{S}_{ideal}$ such that $\pi(\mathcal{F}_{A\text{-}AC}||\mathcal{F}_{IB}) = \mathcal{S}_{real}$, and $\mathcal{S}_{ideal} = \sigma^A \mathcal{F}_{A\text{-}EA}$. Then we prove that $\Delta^{\mathsf{D}}[\mathcal{S}_{real}, \mathcal{S}_{ideal}] \leq \frac{q_s(q_s+q_v)}{|\mathcal{M}|}$.

We notice that $\mathcal{S}_{real}$ and $\mathcal{S}_{ideal}$ behave the same as long as rc generated is different from all the messages authenticated so far and the adversary does not guess rc ahead of time. If any of these cases do not happen then $\mathcal{S}_{real}$ may allow the adversary to authenticate and $\mathcal{S}_{ideal}$ will not. The probability that the server produces two equal random challenges is at most $\frac{q_s^2}{|\mathcal{M}|}$ and the probability that the adversary guesses a random challenge ahead of time is at most $\frac{q_v q_s}{|\mathcal{M}|}$. All that is left is to describe the simulator σ, $\mathcal{S}_{real}$ and $\mathcal{S}_{ideal}$ which is done below. □

The Simulator

A detailed description of the simulator σ can be found in Figure 18. The main goal of the simulator is to maintain a set G message counters (identifying responses from clients) which can be used to succesfully authenticate in the current session.[27]

Defining $\mathcal{S}_{real}$ and $\mathcal{S}_{ideal}$

In this section we describe how systems real and ideal behave when seen as single monolithic systems with $n + 2$ interfaces $C_1, \ldots, C_n$, A and S i.e. from

[27] In particular once an adversary makes an authentication attempt the set G may be cleared since at most one attempt can be made per session. However this behaviour is already enforced by the server protocol in the real world and by the $\mathcal{F}_{A\text{-}EA}$ functionality in the ideal world. Thus clearing G while technically correct would be redundant.

the point of view of the distinguisher. In other words in Figure 19 we describe the system $\mathcal{S}_{real}$ which is obtained by composing all protocols and resources of $\pi^{C_1} \cdots \pi^{C_n} \rho^S(\mathcal{F}_{A\text{-}AC} || \mathcal{F}_{IB})$ (and simplifying some redundant logic as described below). Similarly in Figure 20 we describe $\mathcal{S}_{ideal}$ which captures the behavior of $\sigma^A(\mathcal{F}_{A\text{-}EA})$.

INIT: $rc \leftarrow \perp, c \leftarrow \perp, G \leftarrow \emptyset$
ON (INNER) INTERFACE F:
  CASE (`GO`): $G \leftarrow \emptyset$; Sample $rc \leftarrow_R \mathcal{M}$; output rc on interface A
  CASE (j):
    If ($c = rc$) then $G \leftarrow G \cup \{j\}$
    Output (j, c) on interface A
ON (OUTER) INTERFACE A:
  CASE (m): $c \leftarrow m$, output `QUERY` on interface F
  CASE (j): If ($j \in G$) then output j on interface F, otherwise output $\perp$ on interface F

Fig. 18. The simulator σ

INIT: $rc \leftarrow \perp$, $\forall i \in [n]$ $buffer_i \leftarrow \perp$, $counter \leftarrow 0, M \leftarrow \emptyset$
ON INTERFACE C_i:
  CASE (`RESPOND`):
    If ($buffer_i \neq \perp$) then
      $counter \leftarrow counter + 1$
      $M \leftarrow M \cup \{(counter, i, buffer_i)\}$
      output $(counter, buffer_i)$ on interface A and set $buffer_i$ to $\perp$
ON INTERFACE S:
  CASE (`GO`): Output $rc \leftarrow_R \mathcal{M}$ on interface A
ON INTERFACE A:
  CASE (m): For all $i \in [n]$ $buffer_i \leftarrow m$
  CASE ($j \in \mathbb{N} \cup \{\perp\}$):
    If ($rc \neq \perp$) then
      If ($\exists i$ $(j, i, rc) \in M$) then output i on interface S
      Otherwise output $\perp$ on interface S
      $rc \leftarrow \perp$

Fig. 19. The essential functionality $\mathcal{S}_{real}$ of $\mathcal{F}_{A\text{-}AC}||\mathcal{F}_{IB}$ with π_i's and ρ

INIT: $rc \leftarrow \perp, c \leftarrow \perp, InSession \leftarrow$ `false`, $counter \leftarrow 0$, $Compromised \leftarrow \emptyset$, $\forall i \in [n]$ $msg_i \leftarrow$ `false`
ON INTERFACE C_i:
  CASE (`RESPOND`):
    If ($msg_i =$ `true`) then
      $msg_i \leftarrow$ `false`, $counter \leftarrow counter + 1$
      If ($InSession =$ `true` and $rc = c$) then $Compromised \leftarrow Compromised \cup \{(counter, i)\}$
      Output $(counter, c)$ on interface A
ON INTERFACE S:
  CASE (`GO`):
    $InSession \leftarrow$ `true`, $Compromised \leftarrow \emptyset$
    Sample $rc \leftarrow_R \mathcal{M}$; output rc on interface A
ON INTERFACE A:
  CASE (m): $c \leftarrow m$; $\forall i \in [n]$ $msg_i \leftarrow$ `true`
  CASE ($j \in \mathbb{N} \cup \{\perp\}$):
    If ($InSession =$ `true`) then
      $InSession \leftarrow$ `false`
      If $\exists (j, i) \in Compromised$ then output i on interface S
      Else output $\perp$ on interface S

Fig. 20. The essential functionality $\mathcal{S}_{ideal}$ of $\mathcal{F}_{A\text{-}EA}$ with σ