Unconditionally Secure Ring Authentication

Reihaneh Safavi-Naini
University of Wollongong
5 Northfields Avenue
NSW 2522, Australia
rei@uow.edu.au

Shuhong Wang* †
University of Wollongong
5 Northfields Avenue
NSW 2522, Australia
shuhong@uow.edu.au

Yvo Desmedt
University College London
Gower Street
London WC1E 6BT, UK
y.desmedt@cs.ucl.ac.uk

ABSTRACT

We propose ring authentication in unconditionally secure setting. In a ring authentication system a sender can choose a set of users and construct an authenticated message for a receiver such that the receiver can verify authenticity of the message with respect to the user group chosen by the real sender. The sender will be unconditionally secure even if the receiver has corrupted up to c users and has access to up to ℓ past messages in the system. This functionality is similar to the one provided by ring signature systems with the difference that protection is against an adversary with unlimited power. (This also implies that the verification is not public and is by group members.) In ring signatures an adversary with unlimited computational power can always forge signed messages attributing them to groups of his choice. In our proposed systems the success chance of the adversary can be reduced to the required security of the system. We define model, propose a generic construction whose security is reduced to the security of its building blocks, and give concrete examples of this construction. The construction can also be used in computational setting resulting in ring authentication systems without public key cryptography.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Security

Keywords

Authentication codes, Unconditional security, Ring signature

*The corresponding author.

†This work is in part supported by the Australian Research Council, Discovery Project grant DP0558490.

ASIACCS'07, March 20-22, 2007, Singapore.

1. INTRODUCTION

Consider the following scenario. Alice and Bob are members of a group of users. Group members are connected via authenticated channels. Each user has a shared key with every other user that allows him to authenticate the communication between them. Messages sent in the system have a header that includes the sender and receiver's identities, allowing the receiver to know if a message is aimed at him and verify authenticity of the message using the shared key with the named sender. Alice is a privacy conscious user who requires her communication not to be directly attributable to her. She wants Bob to be convinced about the integrity of the message, but only know that the message is coming from a group of potential senders (that includes Alice) and not be able to point to Alice as the sender of the message.

A similar scenario was considered by Rivest, Shamir and Tauman [24] in computational setting and assuming each user has a (certified) public key. They proposed a *ring signature* that allows a user u to choose a set X of users, each with a public key, and construct a signed message that is publicly verifiable using the set of public keys of members of X hence implying that the signer is a member of X. Ring signatures provide the highest level of anonymity (unconditional and un-revocable anonymity) for senders. However authentication guarantee is computational and an unbounded adversary can always forge signed messages that are attributable to any user group of his choice and in particular to groups that he is not a member of.

Our aim is to provide a functionality similar to ring signature in unconditionally secure setting. That is the aim is to provide anonymity and authenticity of communication against an adversary without requiring any computational assumption. It has been shown that the two most commonly used hard problems in cryptography, factorization and discrete logarithm, have efficient solution if quantum computers are built [28, 29] and so developing systems that remain secure without requiring any hardness assumptions is an attractive research direction.

Similar to ring signatures we allow users to 'hide' themselves in groups chosen by themselves and use group keys to construct authenticated messages that are verifiable by the receiver. Our proposed solution is generic and provides unconditional or computational security depending on the types of primitives used.

1.1 Related Work

Chaum [4] proposed an elegant protocol that provides sender anonymity in an unconditional setting. To our knowl-

edge this is the only protocol providing unconditional sender anonymity without requiring computational assumptions. Dinning Cryptographer (DC) nets allow one member of a user group to send a message that every other group member can decrypt, without origin of the message to be known. The protocol however incurs high communication overhead: for a message of length n in a group of size N, nN bits must be sent. Also other issues such as the need for synchronization and collaboration of all group members must be addressed. In DC nets it is assumed that all group members are honest and there is no protection against tampering with messages: in fact if an adversary controls one of the users, he can easily add an arbitrary string to the message broadcasted by that user, hence modifying the decoded message without being detectable.

Anonymity sets, introduced by Chaum, give the set of potential senders for a message and are used to describe uncertainty about the sender of a message. Serjantov and Danezis [27] argued that membership of anonymity set can be probabilistic with members having higher or lower probability of being a sender, and used entropy of anonymity set as a measure for the level of anonymity provided by a system. Probabilities used in this evaluation could be obtained through objective methods of analyzing the system and the communicated messages, and subjective methods of estimating probabilities. In the absence of a reliable probability distribution on the anonymity set, the size of this set has been used to indicate the level of anonymity provided by the system.

In computational setting a number of models for providing anonymous authenticated communication have been proposed. Ring signatures were originally proposed to allow an insider, for example a parliamentarian, to 'leak' a secret [24] without the threat of retribution and allowing the receiver to be sure that the message is in fact from one of the parliamentarians. The schemes have found other applications that use *ad hoc* group formation property and in all cases ensuring authenticity with respect to a set of public keys, instead of a single one. An attractive property of the scheme, at least in its original form, is that it does not require any special set-up and can use an existing public key infrastructure. It also allows the user to choose arbitrarily the size of the anonymity set.

In group signature [5] a group member can sign a message such that it is verifiable using a group public key, hence guaranteeing that the message is generated by one of the group members. In this system users need to join the group and obtain membership certificates before being able to sign a message. Group signatures usually[1] provide revocability: that is the group manager can reveal the identity of the signer of a signed message. This provides a mechanism for ensuring accountability and that the system is not misused. This is an important property of group signatures and has been refined [1, 15] to include finer level of privacy control by separating group manager and the tracer. Group signatures have been used in real life applications including implementation of trusted computing platform [3]. Anonymity set in group signatures is always the whole group and cannot be controlled by the signer.

Another related model is *ad hoc* anonymous identification scheme [10] which allows a user to form an *ad hoc* group and then prove its membership of the group in an anonymous way. Using Fiat-Shamir transform [12], the scheme can be converted to a constant-size ring signature scheme provably secure in the Random Oracle Model.

[1]There are cases that anonymity is unrecoverable [23].

1.2 Our Work

We propose a sender anonymous system without requiring any computational assumption. We assume users use an *authentication code (A-code)* with pre-distributed shared keys to authenticate messages sent between them. A two party A-code is a symmetric key primitive that uses a tagging algorithms $Tg(k, \cdot)$ to construct a cryptographic checksum for messages, allowing a verifier with a shared key to verify authenticity of the message. To authenticate a message m a tag $t = Tg(k, m)$ is calculated and appended to m to form $m.t$, where '.' denotes concatenation. For a received pair $m.x$ the receiver checks if $Tg(k, m) = x$ and if true accepts m as authentic.

In the group setting, any pair of users A and B share a key k_{AB}. If A wants to send an authentic message m to B, she first uses the A-code and the shared key k_{AB} to construct a tag $t = Tg(k_{AB}, m)$ and then adds $[A, B]$ as the header to obtain $[A.B].m.t$, which is sent to B. Bob can use the header information to determine the 'claimed sender' and the key k_{AB} that must be used for verification. Bob will accept m as authentic if the verification succeeds under the key k_{AB}.

To provide sender anonymity we enable users to construct authenticated messages that can be guaranteed to have come from a member of a group. The group can be chosen by the sender in an *ad hoc* way allowing him to 'hide' himself among a group of his choice. The sender does not need any interaction with others and can construct the message using his key information and the public information of the system. The message will be verifiable by any other member of the user group that had authenticated channel with others.

This property is similar to ring signatures with the difference that the message is only verifiable by the receiver that the sender chooses and all members of the ad hoc group. This is to be expected in unconditional setting as it is known [25] that one way functions are necessary and sufficient for digital signatures (public verifiability).

In an *unconditionally secure ring authentication system (USRA system)* a user can choose members of an *ad hoc* group (anonymity set) and then construct an authenticated message that is verifiable by the receiver of his choice. The adversary can corrupt up to c users obtaining their secret keys. The security goal of the system is to provide protection against *spoofing* attacks by the adversary who after observing multiple authenticated messages, would like to construct a message that is acceptable by a receiver (possibly different from those of the intercepted messages). This property is analogous to the security property of *unforgeability* in ring signatures. The system must also guarantee that no information other than membership of the anonymity set is leaked about the sender.

We evaluate the security of the system by the success chance of the adversary with access to the key information of a group of users $\mathcal{C}$ in constructing an authenticated message that is acceptable by a receiver j and is originated by an anonymity set X that together with j is disjoint from $\mathcal{C}$, that is $\mathcal{C} \cap (X \cup j) = \emptyset$. A secure USRA ensures that a successful forgery by $\mathcal{C}$ will include a colluder in $\mathcal{C}$ with

a very high probability. This definition is in line with ring signature schemes where an adversary that has the secret key information of a member x of a ring X can always forge a message attributable to X.

We allow the adversary to have access to ℓ past communications in the system. In a system that provides security for ℓ messages, users can send up to ℓ ring authenticated messages. A stronger attacker with access to authentication and verification oracle has been proposed for A-codes [26]. Although, similar to earlier work in authentication [30, 9], we focus on an adversary with access to past communication, our approach and constructions can be extended to support security against this stronger attacker.

Efficiency of the system is measured in terms of the key requirement of users and the extra information that must be appended to a message (header and tag). The user key information is pre-distributed and grows with the level of anonymity required. The information sent over the channel consists of the to-be-authenticated message together with the header information that is proportional to the size of the anonymity set, and the cryptographic checksum that is related to the security level of the system.

We give a generic construction for USRA that uses two building blocks: a non-interactive dynamic conference key distribution protocol (referred as *conference key distribution protocol* hereinafter, or CKD for short) and an authentication code (A-code for short).

In an (ω, c)-CKD protocol users receive some initial secret information from a trusted authority. After this initial phase a user can select a group Ω of ω users that includes himself, and calculate a key that can also be calculated by all members of the group Ω. This allows members of Ω to form a secure conference without the need to interact with others. The system guarantees that an adversary that has corrupted c users and has access to their key information cannot learn anything about the secret key of the conference where none of the colluders is a member of.

In our generic construction a user chooses an anonymity set X that the user is a member of and uses an (ω, c)-CKD protocol to generate a key that is shared among members of X, and the receiver. The calculated conference key is used to select a key for an A-code and to authenticate messages. That the conference key can be generated by any member of X makes the user indistinguishable from members of the anonymity set.

The key can also be calculated by the receiver allowing him to verify authenticity of the message. This also means that the *authentication is deniable*.

We define security of ring authentication systems and give necessary conditions to reduce security of the ring authentication system to the security of the underlying CKD protocol and A-code. In particular a perfectly secure conference key system and a perfectly secure authentication code result in a perfectly secure ring authentication.

We consider two constructions of ring authentication systems using two concrete constructions for CKD protocols. The first CKD protocol is a polynomial based construction introduced in from [2] with a slight extension. Blundo *et al* [2] gave a bound on the key size of perfectly secure CKD protocol and proved their construction satisfies the bound with equality (optimal). The keys generated in this CKD protocol are elements of a finite field and can be used to generate keys in a number of authentication systems. Note that it is always possible to define a mapping from the key space of the CKD to an arbitrary A-code but this may result in a loss of security or efficiency. In case that the key sets of the CKD and the A-code are of the same size, a one-to-one mapping can be used. The second construction uses a special combinatorial structure called *generalized cover free families* [34]. A generalized (ω, c) cover free family ((ω, c)-CFF) is a set system $(\mathcal{X}, \mathcal{B})$ where $\mathcal{X}$ is a set of points and $\mathcal{B}$ is a collection of blocks, each a subset of $\mathcal{X}$. The generalized cover free property guarantees that union of up to c blocks in $\mathcal{B}$ cannot cover intersection of ω blocks. Generalised CFF can be used to construct CKD protocol. In this case the conference key is a subset of elements of X. We show how this set can be mapped into the key set of an A-code to construct a ring authentication system.

For both constructions we prove that with appropriate choices, security of the resulting ring authentication systems are guaranteed by the security of the underlying primitives.

1.2.1 Extensions

We discuss two extensions for our work. Firstly, we would like to remove restriction on the size of anonymity groups and colluding sets. That is, give full flexibility to users to choose the anonymity set and ensure that a colluding group of unrestricted size will not be able to identify the sender. We will achieve these properties using a CKD protocol proposed in [8]. The cost of this extreme level of security and user control is that the key size of users, although proven optimal, will be exponential in the size of the group.

We also consider computational model. Users are assumed to be connected through symmetric authenticated channels using MACs and the aim is to provide user controlled anonymity. To this end, in the generic construction the A-code is replaced by a computationally secure message authentication code. In particular we may use the computationally secure MAC that is obtained from Wegman and Carter [35] construction by replacing the random number generator with a pseudo-random number generator. We justify security of the construction and leave formal analysis and proof for future work. As mentioned earlier in our construction the receiver is not cryptographically distinguishable from the sender group and so the system provides deniable authentication. That is the authenticated message could have been generated by the receiver. In computational setting this will provide a similar functionality to the deniable ring authentication of Noar [22].

The paper is organized as follows. In Section 2 we review some primitives including unconditionally secure A-code, unconditionally secure noninteractive conference key distribution protocol and a combinatorial structure, cover-free family, that are used in the rest of the paper. In Section 3 we provide the model and definitions of USRA system. We then give our generic construction and prove its security in Section 4, followed by two concrete constructions in Section 5. In Section 5.3 we discuss extensions and future work. Finally we conclude in Section 6.

2. PRELIMINARIES

2.1 Authentication Code

Unconditionally secure message authentication systems are realized by authentication codes (A-codes).

For simplicity, we focus on *A-codes without secrecy* in systematic form although our results can be easily applied to A-codes with secrecy.

A systematic A-code without secrecy is a symmetric key authentication system that can be represented by a four-tuple $(\mathcal{M}, \mathcal{E}, \mathcal{T}, Tg)$ where $\mathcal{M}$,$\mathcal{E}$ and $\mathcal{T}$ are sets of messages, encoding rules, and tags, respectively. Tg is a tagging algorithm $Tg : \mathcal{E} \times \mathcal{M} \to \mathcal{T}$. To construct an authenticated message for m using the encoding rule $e \in \mathcal{E}$, the tagging algorithm is used to generate a tag $t = Tg(e, m)$ that will be appended to the message and forms $m.t$ where '.' denote concatenation. To verify a tagged message $m.x$ under the rule e, the verifier examines whether $x = Tg(e, m)$, and if true accepts m as authentic. The tagging algorithm can be represented by an $|\mathcal{E}| \times |\mathcal{M}|$ matrix $\mathtt{A}$, whose rows are indexed by encoding rules in $\mathcal{E}$ and columns by messages in $\mathcal{M}$ to be authenticated. The entry $\mathtt{A}(e, m)$ is the $Tg(e, m)$ which is an element t of the tag set $\mathcal{T}$.

An A-code provides security against an adversary who does not have access to the secret key and has observed past authenticated messages and attempts to construct a forgery that will be hopefully acceptable by the verifier. In traditional *spoofing of order* ℓ attack the forgery is constructed after the adversary has observed ℓ authenticated messages $m_1.t_1$, $m_2.t_2$, $\cdots$, $m_\ell.t_\ell$, all generated using the same key. *Impersonation* and *substitution* are spoofing of order zero and one respectively. Let $P_\ell^{A\text{-}code}$ denote the highest success probability of an adversary in an spoofing of order ℓ attack. That is $P_\ell^{A\text{-}code} =$

$$\max_{e} \max_{strategies} p^{A\text{-}code}[m.t \text{ valid for } e | m.t, m_1.t_1, \cdots, m_\ell.t_\ell]$$

where the second maximum is over all the strategies of the adversary in choosing $m.t \neq m_i.t_i$ for all $i \in [\ell]$.

It is proved that for A-codes without secrecy $P_\ell^{A\text{-}code} \geq 1/|\mathcal{T}|$. Codes with *perfect protection of order* ℓ satisfy this bound with equality. An A-code is said to provide ϵ-security of order ℓ if $P_l^{A\text{-}code} \leq \epsilon, 0 \leq l \leq \ell$. A stronger attack model for A-codes has been proposed in [26] where the adversary has access to authentication and verification oracles. In this paper we focus on the traditional attack model but our results can be extended to the stronger attack model.

A widely used construction for A-codes is $(\mathcal{M}, \mathcal{E}, \mathcal{T}, Tg) = (\mathbb{F}_q, \mathbb{F}_q \times \mathbb{F}_q, \mathbb{F}_q, Tg)$ where for $e = (a, b) \in \mathbb{F}_q^2$ we have $Tg(e, m) = a \times m + b$ and all operations are in the finite field $\mathbb{F}_q$. The construction provides perfect protection for spoofing of order up to one but is completely insecure for authentication of more than one messages. Wegman and Carter [35] showed a very efficient construction for authenticating multiple messages. The construction uses almost universal hash families and a one-time-pad. A ϵ-ASU$(n; a, b)$ *ϵ-almost-strongly-universal* hash family is a family of n functions from an a-set A to an b-set B satisfying the following two properties: (i) for any $x \in A$ and $y \in B$, there are exactly n/b functions f such that $f(x) = y$; and (ii) for any two distinct elements x_1 and x_2 in A and two arbitrary, not necessarily distinct, elements of B, there are at most $\epsilon n/b$ functions f such that $f(x_1) = y_1$ and $f(x_2) = y_2$.

In Wegman-Carter construction, to authenticate a sequence of at most ℓ messages a key e is used to determine a function h_e from an ϵ-ASU hash family together with ℓ random tags $b_1 \cdots b_\ell$ (a sequence of ℓ one-time tags). Each message has a counter i attached to it. The tag for the l^{th} message is $Tg(e, m, l) = f_e(m) + b_l$. The construction guarantees $P_l \leq \epsilon, l = 1, \cdots \ell$. Wegman-Carter construction has been used to construct computationally secure MACs such as UMAC [17].

2.2 Conference Key Distribution

Non-interactive dynamic conference key distribution schemes providing unconditional security were introduced by Blundo *et al* [2] and further studied by Desmedt *et al* [8]. They can briefly described as follows.

There is a set of users $\mathcal{U} = \{u_1 \cdots u_N\}$. The aim of the system is to enable members of a conference of c users to individually calculate a secure key that cannot be computed by those who are not in the conference. The system consists of the following phases.

1. *Initialization* A trusted initializer (TI) distributes an initial secret $s_x \in \mathcal{IK}$ (for some initial key domain $\mathcal{IK}$) to each user u_x. Then TI stays offline and does not participate in the protocol.
2. *Conference Key Generation* When a user u_i wants to hold a conference X of users that includes himself, he uses his initial secret and the public values of the system and other users to non-interactively calculate a key k_X using a *key calculation function*.

Let g denote the function that is used to compute the key. Then g satisfies $g(s_x, X) = g(s_y, X) (= k_X \in \mathcal{K})$ for all $x, y \in X$ and so the conference key can also be computed by all other members of X.

Let ω and c be non-negative integers with $\omega + c \leq N$, the total number of users. A non-interactive c-secure ω-conference key distribution protocol, denoted by (ω, c)-CKD, is a two-phase protocol $(\mathcal{U}, \mathcal{IK}, \mathcal{K}, g)$ as described above such that,

1. *Each member of a group of ω users can individually and non-interactively compute a common key.*
   That is, for all $X \in [N] = \{1, 2, \cdots, N\}$ of size ω and for each user u_x with $x \in X$, a unique $k_X \in \mathcal{K}$ exists such that $p^{Conf}[k_X | s_x] = 1$. The superscript $Conf$ stands for the CKD scheme.
2. *Any group of c users have no information on any key they should not know.*
   That is, for all $X, Y \subset [N]$ with $|X| = \omega, |Y| = c$ and $X \cap Y = \emptyset$, it holds that $p^{Conf}[k_X | s_Y] = p^{Conf}[k_X]$, where $p^{Conf}[k_X]$ denotes the *a priori* probability of the conference key being k_X for a given X.

Since each user u_x deterministically computes the conference key k_X from the information s_x received from the server, the probability distribution on $s_X = (s_{i_1}, s_{i_2}, \cdots, s_{i_\omega})$ naturally induces a probability distribution on k_X. Property 2 says that random variables of k_X and of s_Y are statistically independent, thus the information held by U_Y reveals no information on the conference key for U_X unless $X \cap Y \neq \emptyset$.

Obviously a (ω, c)-CKD protocol is also a (ω, c')-CKD protocol for every $c' \leq c$.

THEOREM 2.1. *(Lemma 3.1 [2]) Let $X, X_1, \cdots, X_l, \mathcal{C} \subset [n]$ such that $|\mathcal{C}| = c$, $\mathcal{C} \cap X = \emptyset, \mathcal{C} \cap X_i \neq \emptyset$ and $|X| = |X_i| = \omega$ for $i = 1, 2, \cdots, l$. Then in any (ω, c)-CKD protocol, it holds that*

$$H(S_X | S_{X_1}, \cdots, S_{X_l}) = H(S_X).$$

An (ω, c)-CKD protocol is said to be *perfectly secure* if $p^{Conf}[k_X|s_Y; k_{X_1}, \cdots, k_{X_r}] = p^{Conf}[k_X]$ holds for all X, X_i, Y and r with $|X| = |X_i| = \omega, |Y| = c$ and $X \cap Y = \emptyset$.

2.3 Cover-free Family

Cover-free families (CFFs) are extensively studied set systems which were first introduced in 1964 by Kautz and Singleton [14] in the context of superimposed binary codes. CFFs have been used to solve some new problems in cryptography and communications, including blacklisting [18], broadcast encryption [13], broadcast anti-jamming [7], source authentication in a network setting [21, 11], and group key pre-distribution [31, 32, 33]. A cover-free family is first of all a set system.

A *set system* is a pair $(\mathcal{X}; \mathcal{B})$ with $\mathcal{X}$ being a set of points and $\mathcal{B}$ being a set of subsets of $\mathcal{X}$. Elements of $\mathcal{X}$ are called points and elements of $\mathcal{B}$ are called blocks. A set system $(\mathcal{X}; \mathcal{B})$ can be represented as an $|\mathcal{B}| \times |\mathcal{X}|$ matrix $\mathtt{I}$ called as *incidence matrix*, where $\mathtt{I}(b, x) = 1$ if block B_b includes point x and zero otherwise.

We recall the general definition of CFF given in [34] below.

DEFINITION 2.1. *Let ω, c, d be positive integers. A set system $(\mathcal{X}; \mathcal{B})$ is called a (ω, c, d)-cover-free family, denoted by (ω, c, d)-CFF), provided that, for any ω blocks $B_1, \cdots, B_\omega \in \mathcal{B}$ and any other c blocks $B'_1, \cdots, B'_c$ in $\mathcal{B}$, one has*

$$\bigcap_{i=1}^{\omega} B_i \setminus \bigcup_{j=1}^{c} B'_j \geq d. \quad (1)$$

Obviously any (ω, c, d)-cover free family is also a (ω, c', d')-CFF for any $c' < c$ and $d' < d$.

3. DEFINITIONS OF RING AUTHENTICATION

Let N be a natural number and the set of users be $\mathcal{U} = \{u_1 \cdots u_N\}$. In a *sender-anonymous authentication system*, a sender u_i can send an authenticated message $m.t$ to a receiver u_j such that the receiver can verify authenticity of the message while neither the receiver nor any other observer can trace the message to a single sender. The *sender-anonymity set* or simply *anonymity set* of the sender for the message is the set (size at least 2) of potential senders of the message. That is from the receiver's view point any member of the set could have sent the message.

A sender-anonymous authentication can be trivially constructed as below.

Trivial System T1: *An A-code $(\mathcal{M}, \mathcal{K}, \mathcal{T}, Tg)$ is used for constructing authenticated messages. A trusted initializer randomly selects a key $k \in \mathcal{K}$ and securely gives the key to all users in $\mathcal{U}$. To authenticate a message m, a user u_i computes the authentication tag as $t = Tg(k, m)$ using the tagging algorithm Tg of the A-code. The message sent to u_j is $u_j.m.t$. If u_j receives an authenticated message of this form, he can verify it using his key information and will be convinced that it is from one of the group members.*

In the above system if the adversary corrupts a single user the security is completely lost and the adversary can successfully forge.

Receiving a message enables the receiver to use his secret and system information to compute a set which we refer to as *recovered anonymity set* $Anon(u_j, m.t)$. We assume that members of $Anon(u_j, m.t)$ have equal chance of being the sender. Let $\mathcal{U}_j$ denote the set of users $\mathcal{U}$ excluding u_j. Then in T1 it is easy to see that for all u_j and all valid message-tag pair $m.t$, we have $Anon(u_j, m.t) = \mathcal{U}_j$ and the set is fixed and cannot be controlled by the sender.

A second trivial system described below allows to have different anonymity set for each message and different receiver. However this set, can be computed as $Anon(u_j, m.t) = \{u_x : k_{x,j} \in \mathcal{E}(m.t), x \neq j\}$ where $\mathcal{E}(m.t)$ denotes the encoding rules validating $m.t$, is deduced by both the message and the receiver and therefore is not controllable by the sender.

Trivial System T2: *Assume any pair of users u_i, u_j have a shared key $k_{i,j}$. Using their shared key u_i can send an authenticated messages to u_j. More precisely, to send a message m to u_j, the sender u_i calculates the tag $t = Tg(k_{i,j}, m)$ which is appended to m. To verify a received message $m.t$, u_j uses all keys $k_{j,x}$ that he shares with $u_x \in \mathcal{U}_j$ and accepts the message as authentic if $Tg(k_{j,x}, m) = t$ for some $u_x \in \mathcal{U}_j$.*

We would like to allow senders to determine their anonymity set U_X; here $X \subset [N]$ specifies the indices of users in U_X. This is called *designed anonymity set* where 'designed' informally means that the authenticated messages can only be generated (or verified) by users in $U_X \cup u_j$. T1 and T2 are examples where users cannot control their anonymity set. When the anonymity set can be chosen by the sender, we require the sender to append $[X.j]$ to $m.t$ as a header. So the authenticated message sent over the channel will be $[X.j].m.t$. For an honest sender, a secure sender anonymous system must guarantee that $U_X \subseteq Anon(u_j, m.t)$, where the extra users in $Anon(u_,m.t)$ may be introduced by receiver's inability in calculating the correct anonymity set.

Throughout this paper, we will use *anonymity set* to mean both the recovered anonymity set and the designed anonymity set, expecting that the correct interpretation can be seen from the context. We will use Ω as $[X.j]$ for clear presentation, and do not distinguish it from $X \cup j$, expecting the meaning to be known from the context.

As noted before, in T1 $Anon(u_j, m.t) = \mathcal{U}_j$ and so a received message is attributable to *any* group member. Thus T1 achieves the maximal anonymity. This however results in complete loss of accountability and allowing any group member to easily (no extra) construct messages with no trace to himself.

To provide unconditional anonymity and flexibility for users to choose the anonymity set and provide and authentication guarantee against adversaries who have corrupted a group of users, we introduce *unconditionally secure ring authentication system* (USRA system, for short).

DEFINITION 3.1. *A (r, c, N)-USRA system $(\mathcal{M}, \mathcal{T}, \mathcal{K})$ is a sender-anonymous authentication system consisting of the following three algorithms* $(\mathsf{RAI}, \mathsf{RTg}, \mathsf{RVf})$*:*

1. Initialization $\mathsf{RAI}(r, c, N)$: *The system is initialized by a Trusted Initializer (TI) with parameters r, c, N At the end of running* RAI*, for each $x \in [N]$ a secret $s_x \in \mathcal{K}$ is securely delivered to user u_x .*

2. Ring Authentication Tag $\mathsf{RTg}(s_i, \Omega, m)$: *Takes as input, a secret key s_i for the sender u_i, a designed anonymity*

set (or equivalent information) X, a receiver identity j, and the message $m \in \mathcal{M}$ to be authenticated. RTg *then calculates and outputs a tag $t \in \mathcal{T}$. The authenticated message to be sent to the receiver is $\Omega.m.t$.*

3. Ring Authentication Verification $\mathsf{RVf}(s_j, \Omega.m.t)$: *On receiving a tagged message $\Omega.m.t$ the receiver u_j uses $\mathsf{RVf}(s_j, \Omega.m.t)$ for verification and accepts m as authentic if* RVf *outputs 1 and rejects m otherwise.*

In the following we give definitions of the system parameters and describe the significance.

Correctness: For an (r, c, N)-USRA it is required that for all u_i, u_j, Ω and $m.t$ the algorithm $\mathsf{RVf}(s_j, \Omega.m.t)$ always outputs 1 on messages $\Omega.m.t$ that are correctly constructed as $t = \mathsf{RTg}(s_i, \Omega, m)$. More specifically, for all $i, j \in [N], X \in [N]_r$ with $i \in X$ and for all $m \in \mathcal{M}$, it holds that $\mathsf{RVf}(s_j, \Omega.m.\mathsf{RTg}(s_i, \Omega, m)) = 1$. (Recall that $\Omega = X.j$ and $i \in \mathcal{X}$.)

Anonymity: In general the level of anonymity that a system provides is related to the size of anonymity set. However using set size assumes that the probability distribution on $Anon(u_j, m.t)$ is uniform. More refined measure of anonymity levels can be defined based on the difference between entropy of the a priori distribution on the user set and the distribution calculated after a message is received. We use the size of the designed anonymity set U_X as the measure of anonymity and require $U_X \subseteq Anon(u_j, m.t)$. A similar property holds for ring signatures where the anonymity set is read from the message.

DEFINITION 3.2. *A USRA system is called a* strong (r, c, N) - USRA system *if it is also a (r', c, N)-USRA system for all $r' \leq r$.*

A strong USRA allows users to choose anonymity sets of size up to r, without affecting his security.

DEFINITION 3.3. *An N users USRA system is called* perfect *if it is also a (r, c, N)-USRA system for all $r, c \in [N]$ such that $r + c \leq N - 1$.*

In a perfect system a sender can choose any size for his anonymity set and no colluding group, with no limitation on the size, can have a successful forgery.

Security In an authentication system the main goal of an adversary is to construct a fraudulent message that is acceptable by a receiver. We allow the adversary to corrupt up to c users. We also refer to this as c insiders forming a colluding set $\mathcal{C}$. The aim of the adversary (or colluding group) is to forge a message that is acceptable by a group member. A forgery is considered successful if it is accepted by a receiver and it is from an anonymity set that does not intersect $\mathcal{C}$.

In *spoofing of order ℓ* attack in a USRA system the adversary observes ℓ past authenticated messages, $\Omega_1.m_1.t_1$, $\Omega_2.m_2.t_2, \cdots, \Omega_2.m_\ell.t_\ell$ in the system (remember $\Omega_l = X_l.j_l$). In general messages have different anonymity sets or receivers and hence the tags are constructed using different keys[2]. The adversary constructs a fraudulent message $\Omega.m.t$ and succeeds if $\Omega.m.t$ is accepted by u_j, and attributed to anonymity set U_X.

The best success probability of such order ℓ adversary who has corrupted a set $\mathcal{C}$ of users and knows their key information in spoofing of order ℓ is denoted by P_ℓ^{RA}, where the superscript RA stands for unconditionally secure ring authentication. Then $P_\ell^{RA} =$

$$\max_{all\ strategies} p^{RA}[\mathsf{RVf}(s_j, \Omega.m.t) = 1 | s_{\mathcal{C}}, \Omega_1.m_1.t_1, \cdots, \Omega_\ell.m_\ell.t_\ell]$$

where the set $s_{\mathcal{C}}$ is the key set of the colluders that is accessible to the adversary and the maximum is over strategies of the adversary in choosing Ω and $m.t \neq m_l.t_l$ for all $l \in [\ell]$. Note that the definition includes the case that the messages have the same anonymity set and receiver (i.e., Ω) and hence are constructed under the same key.

DEFINITION 3.4. *A (r, c, N)-USRA system is said to have* ϵ-security of order ℓ, *if the success probability P_ℓ^{RA} as described above is at most ϵ.*

4. A GENERIC CONSTRUCTION

In this section, we give a generic construction GRA for ring authentication system with unconditional security against up to c colluders. The sender can choose an anonymity set of size up to r. The construction uses an A-code with protection against spoofing of order ℓ attacks and a (ω, c)-CKD protocol for N users, with $\omega = r+1$ and $\omega + c \leq N$, as building blocks. We prove the GRA is a secure (r, c, N)-USRA system where security is as defined in Section 3.

DEFINITION 4.1. *We say an A-code $(\mathcal{M}, \mathcal{E}, \mathcal{T}, Tg)$ and an CKD protocol $(\mathcal{U}, \mathcal{IK}, \mathcal{K}, g)$ are* compatible *if an injection mapping f can be defined from the set $\mathcal{K}$ to the set $\mathcal{E}$.*

The GRA system.

Let the A-code $(\mathcal{M}, \mathcal{E}, \mathcal{T}, Tg)$ and (ω, c)-CKD protocol $(\mathcal{U}, \mathcal{IK}, \mathcal{K}, g)$ be compatible. Then GRA=$(\mathcal{M}, \mathcal{T}, \mathcal{IK})$ constitutes a ring authentication system for user group $\mathcal{U}$ and the algorithms $(\mathsf{GRI}, \mathsf{GTg}, \mathsf{GVf})$ are constructed as follows.

1. $\mathsf{GRI}(\omega - 1, c, N) =$ *the initialization of (ω, c)-CKD).* That is, $\mathsf{GRI}(\omega - 1, c, N)$ runs the *Initialization* algorithm of (ω, c)-CKD protocol. At the end of this phase a user u receives a secret $s \in \mathcal{IK}$.

2. $\mathsf{GTg}(s_i, \Omega, m) = Tg(f \circ g(s_i, \Omega), m)$.
This algorithm is invoked by the sender u_i to generate a tag for m and to construct an authenticated message. If u_i wants to send a message to u_j he will perform the following steps.

   (a) Chooses an anonymity set U_X of size $r = \omega - 1$ such that $i \in X$ but $j \notin X$.

   (b) Sets $\Omega = X \cup j$ and generates a conference key $k_\Omega = g(s_i, \Omega)$, where g is the *key calculation* function of the CKD protocol and s_i is the secret distributed to him by TI in the system initialization phase.

   (c) Computes an A-code rule $e_\Omega = f(k_\Omega)$ using the public mapping f. The tag is calculated as $t = Tg(e_\Omega, m)$ using the tagging function Tg of the A-code.

[2]Otherwise, as pointed out in Section 1.2, mapping function (i.e., f later) may result in a loss of certain security if it is many-to-one.

(d) The final authenticated message is $\Omega.m.t$ (meaning $[X.j].m.t$).

3. $\mathsf{GVf}(s_j, \Omega.m.t) = 1$ *if* $t = Tg(f \circ g(s_j, \Omega), m)$.
The verification algorithm invokes the tag function of the underlying A-code using a key that the receiver calculates based on the prefix information of the received message. If u_j receives $\Omega.m.t$, he does the following.

(a) Reads the message header Ω and rejects if $j \notin \Omega$.

(b) Computes the conference key $k_\Omega = g(s_j, \Omega)$ and the encoding rule $e_X = f(k_\Omega)$.

(c) Accepts m as authentic if $t = Tg(e_\Omega, m)$ and rejects otherwise.

Correctness of the GRA system follows immediately from the correctness of the A-code and the conference key distribution protocol. The required computation is the combination of that required by the A-code and the CKD protocol. An authenticated message is always prefixed by Ω that has length $r \cdot \log n$. Sender-anonymity is unconditionally provided at level r since anyone in U_X is able to construct the authenticated message. The following theorem guarantees the security of the GRA system under spoofing attacks given the underlying primitives are both secure in respective terms.

THEOREM 4.1. *In the generic construction above, if (i) the A-code has ϵ-security against spoofing attack of order ℓ, and (ii) the (ω, c)-CKD protocol has perfect security, then the GRA system is ϵ-secure against spoofing attack of order ℓ, assuming the adversary is allowed to corrupt up to c users.*

PROOF. (*sketch*) We only prove the theorem for $\ell = 1$. The proof can be extended to the general case with $\ell > 1$. We relate the best success probability of the adversary in GRA system to the success probability of an attacker in the underlying A-code, using the perfect security of the CKD.

The success probability of a GRA adversary with a tagged message $\Omega.m.t$ after observing an authenticated message $\Omega_1.m_1.t_1$ is denoted by $p^{GRA}[m.t \text{ is valid for } \Omega | s_{\mathcal{C}}; \Omega_1.m_1.t_1]$. Let P_1^{GRA} denote the best success probability of the GRA adversary after observing one message. This is given by,

$$P_1^{GRA} = \max_{\Omega.m.t | \Omega_1.m_1.t_1} p^{GRA}[\Omega.m.t \text{ is valid} | s_{\mathcal{C}}; \Omega_1.m_1.t_1]$$

$$= \max_{\Omega_1.m_1.t_1} \max_{\Omega.m.t} p^{GRA}[\Omega.m.t \text{ is valid} | s_{\mathcal{C}}; \Omega_1.m_1.t_1]$$

The A-code adversary's success probability with a tagged message $m.t$ after observing an authenticated message $m_1.t_1$ is given by $p^{A\text{-}code}[m.t \text{ is valid} | m_1.t_1]$ and its best success probability is $P_1^{A\text{-}code} = \max_{m.t | m_1.t_1}[m.t \text{ is valid}]$.

Given a GRA adversary who uses $\Omega.m.t$ after observing $\Omega_1.m_1.t_1$, we construct an A-code adversary who is given all the key information that the GRA adversary has, and uses $m.t$ as the spoofing message for the A-code after observing $m_1.t_1$.

For any $m_1.t_1$, Ω_1, Ω and $\mathcal{C}$ such that $\mathcal{C} \cap (\Omega \cup \Omega_1) = \emptyset$, the best success probability of the GRA adversary is given by

$$P^{GRA}(s_{\mathcal{C}}; \Omega_1.m_1.t_1)$$
$$= \max_{\Omega.m.t \neq \Omega_1.m_1.t_1} p^{GRA}[m.t \text{ is valid for } \Omega | s_{\mathcal{C}}; \Omega_1.m_1.t_1]$$
$$= \max_{\Omega.m.t \neq \Omega_1.m_1.t_1} p^{GRA}[\mathsf{GVf}(s_j, \Omega.m.t) = 1 | s_{\mathcal{C}}; \Omega_1.m_1.t_1]$$

Now consider two cases. First, let $\Omega_1 = \Omega$. Then, for the A-code we have $Tg(k_\Omega, m_1) = t_1$. The A-code adversary has $s_{\mathcal{C}}$ which because of the perfect security of CKD gives no information about k_Ω. Hence,

$$\max_{m.t \neq m_1.t_1} p^{A\text{-}code}[Tg(e_\Omega, m) = t | s_{\mathcal{C}}; \Omega_1.m_1.t_1]$$
$$= \max_{m.t \neq m_1.t_1} p^{A\text{-}code}[Tg(e_{\Omega_1}, m) = t | Tg(e_{\Omega_1}, m_1) = t_1]$$
$$= P_1^{A\text{-}code}(\Omega_1.m_1.t_1)$$

Second let $\Omega \neq \Omega_1$. Using the same argument,

$$\max_{m.t \neq m_1.t_1} p^{A\text{-}code}[Tg(e_\Omega, m) = t | s_{\mathcal{C}}; \Omega_1.m_1.t_1]$$
$$= \max_{m.t \neq m_1.t_1} p^{A\text{-}code}[Tg(e_\Omega, m) = t]$$
$$\leq P_0^{A\text{-}code}$$

Together

$$P^{GRA}(s_{\mathcal{C}}; \Omega_1.m_1.t_1) \leq \max\{P_0^{A\text{-}code}, P_1^{A\text{-}code}(\Omega_1.m_1.t_1)\}$$

and finally

$$P_1^{GRA} = \max_{\Omega_1.m_1.t_1} P^{GRA}(s_{\mathcal{C}}; \Omega_1.m_1.t_1)$$
$$\leq \max_{\Omega_1.m_1.t_1} \{P_0^{A\text{-}code}, P_1^{A\text{-}code}(\Omega_1.m_1.t_1)\}$$
$$= \{P_0^{A\text{-}code}, P_1^{A\text{-}code}\} \leq \epsilon.$$

5. CONCRETE CONSTRUCTIONS

In this section we show two constructions and discuss on some extensions for ring authentication based on the generic construction.

5.1 Using Polynomial CKD Protocol

We choose the (ω, c)-CKD protocol to be the polynomial scheme in [2]. The protocol allows dynamic conferences to calculate a conference key that is an element of $\mathbb{F}_q$. We use two independent (random choices) copies of the protocol to allow users to calculate a pair $(a, b) \in \mathbb{F}_q^2$ that is used as the key for the authentication system GRA in Section 4. The protocol works as follows.

During the *initialization* phase, the trusted initializer (off-line server) randomly chooses two symmetric polynomials[3] F, G of degree c and in ω variables, with coefficients randomly chosen from a finite field $\mathbb{F}_q$. The initial secret of user u_i is evaluations of F, G at $x_1 = i$. That is, $s_i = (a_i(x_2 \cdots x_\omega), b_i(x_2 \cdots x_\omega))$, where $a_i(x_2 \cdots x_\omega) = F_i(x_2 \cdots x_\omega) = F(i, x_2, \cdots x_\omega)$ and $b_i(x_2 \cdots x_\omega) = G_i(x_2 \cdots x_\omega) = G(i, x_2, \cdots x_\omega)$. Each polynomial will have $\binom{c+\omega-1}{c-1}$ coefficients and so the total key size of a user is $\binom{c+\omega-1}{c-1}$ field elements.

In the *conference key generation* phase, suppose user u_i wants to obtain a common key k_X shared among U_X with

[3]A polynomial $F(x_1 \cdots x_\omega)$ is symmetric if $F(x_1 \cdots x_\omega) = F(x_{\sigma(1)} \cdots x_{\sigma(\omega)})$ for all permutation σ of ω elements.

$X = \{i, i_2, \cdots, i_\omega\} \subset [N]$. Then he computes k_X as the pair $(a_X, b_X) = (F_i(i_2, \cdots, i_\omega), G_i(i_2, \cdots, i_\omega))$.

Correctness and the security of the key pair follows from the security of the original scheme and noting that the two copies of the original CKD protocol were independently chosen.

We choose the A-code described in Section 2.1. For this A-code the set of encoding rules is $\mathcal{E} = \mathbb{F}_q \times \mathbb{F}_q$, the message space is $\mathbb{F}_q$ and tagging function is $Tg((a,b),m) = a \times m + b$.

The code is compatible with the polynomial CKD protocol (2 copies) where the function f is the identical mapping from $\mathbb{F}_q^2 \to \mathbb{F}_q^2$.

Security of the combined scheme follows from Theorem 4.1. The combination results in a $1/q$-secure (r,c)-USRA system with $r = \omega - 1$ where q is the size of $\mathbb{F}_q$.

It is worth to point out that this construction is optimal in term of the key storage needed by the user. This is because the base A-code is optimal – the size of encoding rules is as small as possible, the underlying CKD is also optimal – the size of information held by users is as small as possible, and the function f is an identical mapping.

5.2 Using Generalized Cover-free Family

In this subsection, we use a generalized CFF for constructing the conference key distribution. This construction is a natural extension of other works [31, 32] on key predistribution schemes.

For an N users group, we require the (ω, c, d)-CFF $(d \geq 1)$ $(\mathcal{X}; \mathcal{B})$ to have at least N blocks, i.e., $|\mathcal{B}| \geq N$. Each user is associated with a distinct B. This is possible because $|\mathcal{B}| \geq N$. Let u_i be associated to B_i with blocks ordered arbitrarily.

Suppose there is a $(\mathcal{M}, \mathcal{E}, \mathcal{T}, Tg)$ A-code with ϵ-security against spoofing of order 1 and $|\mathcal{E}| = |\mathcal{X}|$ (the result can be generalized to higher order spoofing in a straightforward way).

In each running of the system, the TI randomly associates one key from $\mathcal{E}$ to each point in $\mathcal{X}$. Without loss of generality let x_l is associated with e_l for $l \in [|\mathcal{E}|]$. User u_i receives all keys whose corresponding points are in B_i.

To construct an authenticated message for a user u_j, u_i does the following.

1. *Key generation:* Selects an anonymity set $X \subset \mathcal{X}$ that includes himself, and forms $\Omega = X \cup j$. He will then finds $\{x_{i_1} \cdots x_{i_h}\} = \cap_{i \in \Omega} B_i$. Let $k_\Omega = \{e_{i_1}, \cdots e_{i_h}\}$. Note that because of the cover free property, no colluding set C will be able to cover $\{x_{i_1} \cdots x_{i_h}\}$ and so there is at least one key $e_{x_l}, 1 \leq l \leq h$ that is only known by members of Ω.

   Let $\mathcal{E}$ be an Abelian group with group operation $\odot$. The encoding rule used for authentication is calculated as $e_\Omega = \odot e_{l \in [h]} i_l$. An example of A-codes that satisfy this property is the code used in the previous construction.

2. *Tag construction*: The tag for a message m is $t = Tg(e_\Omega, m)$. The message sent to u_j is $\Omega.m.t$.

Correctness of the system follows immediately from the correctness of the conference key distribution and the correctness of the A-code

Security of the system following from observing that members of each conference will have a key that is not known by the adversary (colluders). This means that the conference key as defined above has perfect security and $H(E_\Omega | s_\mathcal{C}) = H(E_\Omega)$ where E_Ω is the random variable associated with e_Ω. Hence the conditions of Theorem 4.1 are satisfied and we will have the following theorem.

THEOREM 5.1. *The construction above, referred as CFF-RA, is a $(\omega-1, c)$-USRA system and is secure against spoofing of order 1, i.e., $P_1^{CFF-RA} \leq 1/q$.*

5.3 Extensions

Ring authentication with perfect anonymity.

The generic construction in Section 4 effectively reduces the construction of a ring authentication system to the construction of a non-interactive dynamic conference key system. The two constructions above the key size of users is $\binom{c+\omega-1}{c-1}$ and so polynomially depends on ω and c. The construction is particularly important because it is proven optimal and ensures that the key sizes of the users is minimum. An important property of the construction is that the key size of the users is independent of the size of the group and only depends on the level of anonymity required by the users and acceptable colluding size. The construction using generalized CFF, although in general non-optimal, provides perfect security with key size of the user independent from the group size. However there is no general expression for the users' key size and this will depend on the actual construction of CFF.

We may remove the restriction on the sizes of the anonymity set and the colluding set by allowing users to have larger key sizes. In the scheme proposed in [8] any conference size is permitted and there is no limitation on the size of colluding set. Authors prove that for a group of size N, the sizes of user u_x's secret $H(S_x)$ and of the conference key $H(K)$ satisfy $H(S_x) \geq (2^{N-1} - 1)H(K)$ and propose an optimal system that satisfies the bound with equality. In this optimal system a user receives a key for every conference that he can participate and its key size is $2^{N-1} - 1$. Using this CKD protocol with an A-code guarantees anonymity to an arbitrary level determined by the user.

Efficient constructions for higher level of spoofing.

In our analysis and constructions we assumed the adversary has access to one past authenticated message. Theorem 4.1 can be extended to higher level of spoofing to prove the following result.

$$P_\ell^{RA} = \{P_0^{\text{A-code}}, P_1^{\text{A-code}}, \cdots P_\ell^{\text{A-code}}\}$$

The most efficient construction for A-codes that are secure against spoofing of order ℓ is due to Wegman and Carter [35]. To obtain sufficient key bit for authentication the conference key can be run independently multiple times (similar to the construction in 5.1).

Stronger attacker.

In this paper we focused on message observing adversaries. This is the extensively used adversary model in unconditional setting. Our results however can be generalized to adversaries with oracle access. However the definition of security for ring authentication needs to be extended to include oracle access for the adversary. That will be a future extension of our work.

Computational anonymity.

The generic construction provides a template for providing ring authentication in computational setting. Using a CKD protocol with a computationally secure MAC (e.g. HMAC [16]) provides uncertainty about the sender of the message while allowing receiver to be able to verify authenticity of the message. Modelling and evaluation security in this setting are interesting open questions.

6. CONCLUDING REMARKS

We proposed ring authentication system as a privacy enhancing mechanism for users to hide true sender of a message in a group, referred to as anonymity set.

Users remain unconditionally secure both from the view point of authenticity of communication, and anonymity of sender. Although unconditional sender anonymity is also provided by a ring signature but the traditional ring signature become completely insecure from the view point of the authenticity of communication.

In our model a message has a 'claimed' sender group and is verifiable with respect to that group by a designated receiver. The receiver group can be expanded using a multi-receiver authentication system with dynamic sender [6, 19, 20].

We used the size of anonymity set as the level of anonymity and noted that more refined entropy theoretic measures are possible. An interesting open problem is finding information theoretic bound on the key size of users in a USRA system with perfect security. The construction in Section 5.1 appears optimal because both the underlying primitives are optimal. A further proof of this property requires deriving the information theoretic bound described above.

Extension of ring authentication to computational security will be of practical interest. Ring signature schemes require public key and may exponentiations to sign or verify a message. Using our approach results in a very efficient (computationally) authentication system. However the key size of users grows with the level of anonymity. Assuming a 2048 bit RSA signature, the secret key of a user will be similar to the amount of key material in a ring authentication that allows him to hide himself in a group of 18, using key size 128 bit for the MAC and collusion size 1.

7. ACKNOWLEDGEMENT

We are glad to thank anonymous reviewers for their valuable comments, especially for correcting some typos.

8. REFERENCES

[1] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[2] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly secure key distribution for dynamic conferences. *Inf. Comput.*, 146(1):1–23, 1998.

[3] E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In V. Atluri, B. Pfitzmann, and P. D. McDaniel, editors, *ACM Conference on Computer and Communications Security*, pages 132–145. ACM, 2004.

[4] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[5] D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.

[6] Y. Desmedt, Y. Frankel, and M. Yung. Multi-receiver/multi-sender network security: Efficient authenticated multicast/feedback. In *INFOCOM*, pages 2045–2054, 1992.

[7] Y. Desmedt, R. Safavi-Naini, H. Wang, L. M. Batten, C. Charnes, and J. Pieprzyk. Broadcast anti-jamming systems. *Computer Networks*, 35(2-3):223–236, 2001.

[8] Y. Desmedt and V. Viswanathan. Unconditionally secure dynamic conference key distribution. In *ISIT*, page 383, Cambridge, MA, USA, 1998. IEEE Press.

[9] C. Ding and X. Wang. A coding theory construction of new systematic authentication codes. *Theor. Comput. Sci.*, 330(1):81–99, 2005.

[10] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, 2004.

[11] M. E. Dyer, T. I. Fenner, A. M. Frieze, and A. Thomason. On key storage in secure networks. *J. Cryptology*, 8(4):189–200, 1995.

[12] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[13] J. A. Garay, J. Staddon, and A. Wool. Long-lived broadcast encryption. In M. Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 333–352. Springer, 2000.

[14] W. H. Kautz and R. C. Singleton. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10(4):363–377, 1964.

[15] A. Kiayias and M. Yung. Efficient secure group signatures with dynamic joins and keeping anonymity against group managers. In Dawson and S. Vaudenay, editors, *MYCRYPT*, volume 3715 of *Lecture Notes in Computer Science*, pages 151–170. Springer, 2005.

[16] H. Krawczyk, M. Bellare, and R. Canetti. Hmac: Keyed-hasing for message authentication. IETF RFC 2104, 1997.

[17] T. Krovetz. Software-optimized universal hashing and message authentication. PhD thesis. UC Davis, Department of Computer Science, September 2000.

[18] R. Kumar, S. Rajagopalan, and A. Sahai. Coding constructions for blacklisting problems without computational assumptions. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 609–623. Springer, 1999.

[19] K. Kurosawa and S. Obana. Characterisation of (k, n) multi-receiver authentication. In V. Varadharajan, J. Pieprzyk, and Y. Mu, editors, *ACISP*, volume 1270

of *Lecture Notes in Computer Science*, pages 204–215. Springer, 1997.

[20] K. M. Martin and R. Safavi-Naini. Multisender authentication systems with unconditional security. In Y. Han, T. Okamoto, and S. Qing, editors, *ICICS*, volume 1334 of *Lecture Notes in Computer Science*, pages 130–143. Springer, 1997.

[21] C. Mitchell and F. Piper. Key storage in secure networks. *Discrete Applied Mathematics*, 21(2):215–228, 1988.

[22] M. Naor. Deniable ring authentication. In M. Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 481–498. Springer, 2002.

[23] L. Nguyen and R. Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In P. J. Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 2004.

[24] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[25] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 387–394, New York, NY, USA, 1990. ACM Press.

[26] R. Safavi-Naini and P. Wild. Bounds on authentication systems in query model. In *Theory and Practice in Information-Theoretic Security, ITW'05*, pages 85–91. IEEE Press, 2005.

[27] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In R. Dingledine and P. F. Syverson, editors, *Privacy Enhancing Technologies*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2002.

[28] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[29] P. W. Shor. Quantum computing. In ICM: *Proceedings of the International Congress of Mathematicians*, 1998.

[30] G. J. Simmons. Authentication theory/coding theory. In G. R. Blakley and D. Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 411–431. Springer, 1984.

[31] D. R. Stinson. On some methods for unconditionally secure key distribution and broadcast encryption. *Des. Codes Cryptography*, 12(3):215–243, 1997.

[32] D. R. Stinson and T. van Trung. Some new results on key distribution patterns and broadcast encryption. *Des. Codes Cryptography*, 14(3):261–279, 1998.

[33] D. R. Stinson, T. van Trung, and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *J. Stat. Planning and Inference*, 86(2):595–617, 2000.

[34] D. R. Stinson and R. Wei. Generalized cover-free families. *Discrete Mathematics*, 279(1-3):463–477, 2004.

[35] M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.