General Group Authentication Codes and their Relation to "Unconditionally–Secure Signatures"

Reihaneh Safavi–Naini[1], Luke McAven[1] and Moti Yung[2]

[1] School of Information Technology and Computer Science, University of Wollongong, Wollongong 2522, Australia. [rei,lukemc]@uow.edu.au
[2] Department of Computer Science, Columbia University, New York, NY 10027, USA. moti@cs.columbia.edu

Abstract. Strong notions of security for unconditionally secure digital signature schemes (USDS) were recently proposed where security is defined based on notions of security in computationally–secure digital signatures. The traditional area of unconditionally secure authentication, however, is that of "authentication codes" (A–codes). Relations between primitives is central to cryptographic research. To this end, we develop a novel "general group–based A–code" framework which includes known types of group A–codes and their extensions, including the newly proposed USDS, and also allows other models to be systematically described and analysed. In particular, information theoretic analysis of these codes can be applied to USDS, establishing fundamental bounds on USDS parameters.
A second contribution herein is a modular algebraic method of synthesising group codes from simpler A–codes, such that security of the group code follows directly from the component codes. We demonstrate our approach by constructing and analysing a USDS satisfying the 'strongest security notion'.

1 Introduction

Digital signatures are the basic authentication primitive in modern cryptography. They are known to be equivalent to the existence of one–way functions, and thus to rely on computational assumptions [12]. There are, however, settings where reliance on computational assumptions is inappropriate (typically for small mutually distrusting groups of entities that do not know each others computational or technological advantages, e.g. advances in quantum computations, as is the setting between nations). The alternative model for secure authentication, when there is no assumption regarding adversaries computational power, has been A–codes as suggested by Simmons [16]. This was indeed motivated by authentication procedures between the USA and USSR regarding treaty verification.

In recent years a number of unconditionally secure digital signature schemes, both in interactive [2] and non–interactive settings, have been proposed. We consider a non–interactive setting where a trusted Key Distribution Centres (KDC) or trusted authority (TA) generates and distributes the key information of system participants. The two main approaches satisfying these assumptions are due to Johansson (J99) [11], who considered a variant of multireceiver authentication codes with an untrusted sender and an arbiter, and called it *Unconditionally Secure Digital Signature (USDS)*, and Hanaoka, Shikata, Zheng, and Imai [7, 15] (referred to as HSZI00 and SHZI02, respectively) who recently proposed a range of new security notions for USDS. In Eurocrypt 2002 [15], the authors formalised their approach independent of the theory of A–codes, and proposed the 'strongest notion' of security for USDS without reference to these codes. They constructed a USDS that provided the 'strongest security notion'.

To understand these proposals we develop a unified framework allowing evaluation of USDS schemes within the domain of A–codes. We view the work and scenarios of HSZI00/SHZI02 as providing motivation for studying A–code generalisations. One may mistake the use of new notions in HSZI00/SHZI02 to mean extensions of this theory cannot capture the new settings (and that perhaps a new type of theory, similar to that for conditionally secure signatures, is needed). We believe our work puts things in order, in this respect.

A second contribution of this paper is proposing a modular algebraic method for synthesising group A–codes. This is particularly important because constructing A–codes for complex authentication scenarios can become a formidable task and approaches that allow 're–use' of proven secure schemes as building blocks will provide an attractive option.

Review of *A*–codes

Unconditionally secure authentication codes were first constructed in [6] and then modelled and analysed in [16]. The original A–codes were symmetric key primitives for communication between two honest participants, secure against spoofing attacks. Simmons derived the first information theoretic bound on impersonation, bounds on higher order spoofing were later obtained [10, 18].

Simmons [17] also considered A^2–codes in which sender and receiver are distrusted. The sender may *deny* a sent message, and a receiver may substitute a received message or try to ascribe a message to the sender. A^2–codes are asymmetric primitives in which the sending and receiving keys differ. Simmons showed the need for a trusted *arbiter*, with the key information of the sender and receiver, to resolve disputes. In A^3–codes [1, 3] the trust in the arbiter is reduced and the arbiter may attempt to construct fraudulent messages.

Group–based A–codes were introduced by [4] and extended by [5, 11, 13, 14]. In multireceiver A–codes (MRA) [4] a sender constructs an authenticated message that is verifiable by each member of a verifier group. The sender is trusted but receivers may collude to construct a fraudulent message on behalf of the sender. In (J99) [11] the senders are distrusted, and the resulting system was called an *Unconditionally Secure Digital Signature (USDS)*. In this model the sender may deny his constructed message. We call this model an MRA^2–code, since the trust assumption is most similar to A^2–codes.

Requirements of a USDS:

In a USDS scheme signers and verifiers are distrusted. They may try to forge signed messages, or repudiate their own signed messages. Let $\mathcal{U}$ denote a set of distrusted participants. Any $U_i \in \mathcal{U}$ can sign a message that is verifiable by all $U_j \in \mathcal{U}$. An important property of standard digital signatures is that if U_i obtains a signed message from U_j he can convince U_ℓ that the message is from U_j; this is called transferability. We require the following properties to be satisfied.

Transferability: U_j can convince any $U_k \in \mathcal{U}, k \neq \{i, j\}$, the message is from U_i.
Unforgeability: A colluding subset $C \subset \mathcal{U}$ has a negligible probability of constructing a fraudulent message that is acceptable by a group member U_k as signed by U_i, where:
(i) $U_i \in C$, and can deny the message, (*non–repudiation*).
(ii) $U_i \notin C$, and the message is not generated by U_i.

These properties match the requirements of the first unconditionally secure signature (interactive) protocol [2], and are closest to those achieved in computationally secure signature schemes. An important difference between computationally and unconditionally secure digital signatures is that in USDS verification cannot be a public process, and so secret keys are needed, which, as noted in [11, 13, 15], must be different for each group member.

Our Results:

We propose a common framework for modelling and analysing asymmetric group A–codes and USDS schemes. We introduce *authentication* and *verification oracles* which adversaries interact with to obtain spoofing information. We also introduce *authentication scenarios* and outline a general way of expressing security goals and adversary's power. We give a generalised bound that applies in such scenarios. Our work suggests numerous variations on defining security goals of a group–based A–code and adversaries power. Critically, the framework allows information theoretic *security* and *efficiency* evaluations for USDS.

We also propose a methodical approach to synthesising complex group–based USDS systems with provable security, starting from simple component systems with provable security. This approach is algebraic and while sometimes providing less efficient constructions it avoids some disadvantages of combinatorial synthesis. Furthermore, security proofs follow from security of components.

The rest of the paper is organised as follows: In section 2 we recall parts of A–code theory. In section 3 we propose our model of asymmetric group authentication codes (USDS) and show how previous USDS models fit in this framework. Section 4 contains the new design methodology with concrete constructions, while section 5 sketches our general framework for group authentication. Finally, section 6 contains our concluding comments.

2 Preliminaries

An authentication code may be represented as a 4–tuple, $C = (\mathcal{S}, \mathcal{M}, \mathcal{E}, f)$, where $\mathcal{S}, \mathcal{M}, \mathcal{E}$ are the sets of source states, messages and keys, respectively. The function $f : \mathcal{S} \times \mathcal{E} \to \mathcal{M}$ takes a *source state* s, a key e and generates the corresponding *message* m. The function f defines two algorithms; an *authentication algorithm* used by the sender to generate an authenticated message, and a *verification algorithm* used by the receiver to verify a received message. There is also a *key generation* algorithm that generates key information for the system. We use *systematic Cartesian* A–codes, wherein the messages are of the form (s, t), where the tag t is used to authenticate the source state s. Such an authentication code is represented as a 3–tuple $C = (\mathcal{S}, \mathcal{A}, \mathcal{E})$ with $t = e(s), e \in \mathcal{E}, s \in \mathcal{S}$, and $\mathcal{A}$ being the set of tags (or authenticators). A–codes are symmetric key systems and the secret key is shared by sender and receiver, who are assumed to be trusted.

An attacker may inject a fraudulent message into the system (an *impersonation attack*), or construct a fraudulent message m' after observing a valid message m (a *substitution attack*). In both cases the attacker succeeds if the fraudulent message is accepted. The best success probability of the attacker in the two attacks are denoted P_I and P_S, respectively. A message m is *valid* for a key e if $m \in \mathcal{M}(e)$, where e is the key shared by the sender and receiver. Security of an A–code is defined by the attackers best success probability in the attacks.

$$P_I = \max_{m \in M} p(m \text{ is valid for } e) \qquad P_S = \max_{m' \in M \setminus \{m\}} p(m' \text{ is valid for } e | m) \ .$$

An A–code has ϵ*–security* if the success probability of any attack is at most ϵ.

In A^2–codes one considers signer's *denial attack* and receiver's *impersonation* and *substitution* attacks. In A^3–codes [1, 3] fraud by the arbiter is treated also.

Authentication systems may provide security for more than one message. In *spoofing of order* t, the attackers have access to up to t authenticated messages. Order 0 and 1 spoofing are impersonation and substitution, respectively. Codes that provide security for t–messages are denoted as tA, tA^2 and tA^3–codes.

Efficiency parameters of an A–code include participants key sizes, and the length of the authenticator. Performance bounds provide fundamental limits on these parameters for a given level of security, or alternatively bound the security level for a given set of parameters. Two types of bounds are derived for A–codes: *information theoretic bounds* on the success probability of attacks in terms of information theoretic measures, and *combinatorial bounds* on the sizes of the key spaces and authenticator in the system. Information theoretic bounds for A–codes were given in [6, 16] and later derived for other models [11].

Group–based A–codes (the subject of this work) were introduced in [4] and developed by numerous authors [5, 11, 13, 14]. *Multireceiver A–codes (MRA–codes)* allow a single trusted sender to send a message to a group of receivers such that each receiver can individually verify the message. A (ϵ, w, n)–MRA–code is an MRA–code for which the success probability of the best attack (impersonation and substitution) for a colluding group of w verifiers is less than ϵ. Information theoretic bounds and constructions for such codes are given in [13].

2.1 Constructions:

Numerous constructions of A–codes have been proposed (for example [4, 13, 14, 16]). We briefly recall constructions to be used in this paper.

Polynomial A–code [4] (C_0) Consider the A–code defined by the function $f(x) = a + bx$, where $(a, b) \in \mathbb{F}_q^2$ is the key and the authenticator for the source state $s \in \mathbb{F}_q$ is given by $f(s)$. This code satisfies $P_I = P_S = 1/q$.

Polynomial (ϵ, w, n)–MRA–code [4] (C_1) The sender has two polynomials $f(x)$ and $g(x)$, both of degree at most w, with coefficients over F_q, the finite field with q elements. Each receiver U_i is given $(u_i, f(u_i), g(u_i))$, where $u_i \in \mathbb{F}_q$ is public and $u_i \neq u_j, i \neq j$. To authenticate a source state s, the sender constructs the tag $\alpha(x) = f(x) + sg(x)$ and appends it to s. The receiver U_i accepts a message $(s, \alpha(x))$ as authentic if $f(u_i) + sg(u_i) = \alpha(u_i)$. The construction has $\epsilon = 1/q$ and is *optimal* with respect to tag length, and key sizes.

3 Asymmetric authentication in groups: USDS

We consider systems where no participant is trusted (except, possibly the arbiter), and where participants' keys are only known to themselves, hence the term asymmetric. We focus on single signer schemes.

3.1 A general framework for single signer group A–codes

There is a set $\mathcal{U} = \{U_0, U_1, \ldots, U_n, U_A\}$ of distrusted participants, each with secret key information. The set $\mathcal{U}$ contains n verifiers, an arbiter U_A, and a signer U_0. A message signed by U_0 is acceptable to all verifiers. We assume the arbiter has the algorithm and key information of a verifier, so the arbiter's key information is the same as a verifier's. Arbitration is performed by applying the verification algorithm to a 'suspect' signed message and using the result to resolve the dispute following arbitration rules.

Each user has a distinct identity encoded in the source state: for example the source state can be the concatenation of the user's identity and the information signed. The signer wants to sign $s \in \mathcal{S}$ so any verifier can verify the signature.

The adversary can corrupt a group C, of at most w verifiers, and possibly the signer and/or the arbiter. This is the model in earlier group–based A–codes and USDS. Including the arbiter assesses security under extreme attack conditions. One assumes, however, the arbiter follows the correct arbitration rules.

We consider the following types of attacks.

1. $U_0 \in C$. A *denial attack* where U_0 signs a message, then denies it. Colluders succeed if, following arbitration, the message is deemed not from U_0.
2. $U_0 \notin \mathcal{U}$. In this case the attack is one of the following types.
   - *spoofing attack:* The collusion constructs a message valid for a verifier.
   - *framing attack:* The colluders construct a message attributable to U_0 and acceptable to a verifier. We note the verifier, in this case, may be part of the collusion.

In spoofing attacks colluders succeed if their fraudulent message is acceptable to a target verifier. The message may or may not be valid for (constructible by) U_0.

We remark that HSZI00 introduced an attack against transferability, called 'transfer with a trap'. We show in section 3.2 that this attack has less chance of success than the above attacks and therefore need not be considered separately.

The above requirements are reminiscent of MRA–codes and thus we will use the term MRA^2–codes and MRA^3–codes when the arbiter is, or is not, trusted. With a trusted arbiter, a signer's denial attack succeeds if the colluders construct a message m where $m \notin \mathcal{M}(e_T)$, e_T being the key, e_A the arbiter's key distinct from all e_i, which denote the key of U_i.

We use E_i, E_T, E_A and E_C to denote sets of keys associated with verifier U_i, signer U_0, arbiter U_A, and collusion E_C, respectively. The success in denial attacks can be measured by the probability of a verifier U_i accepting the message, $m \in \mathcal{M}(e_i)$, but the arbiter not, i.e., $m \notin \mathcal{M}(e_A)$. In verifier's spoofing attack the message must be valid for a verifier U_j and so $m \in \mathcal{M}(e_j)$, while in verifier's framing attack $m \in \mathcal{M}(e_T)$ and $m \in \mathcal{M}(e_i)$ for some verifier U_i.

Security of an MRA^2 code against the above attacks can be defined using probabilities, $P_D^{t_{V_1},t_{V_2}}$, $P_{RS}^{t_A,t_{V_1},t_{V_2}}$, and $P_{RS}^{t_A,t_{V_1},t_{V_2}}$. In the first attack the collusion includes the signer, in the last two it does not. Each probability is obtained as the best success probability of colluders. The superscripts represent colluders ability to collect information on uncorrupted verifiers' keys by oracle interaction.

Colluders information

Colluders have their key information. In traditional A–codes colluders may also have access to prior *authenticated messages* sent over the channel. We model such observations by queries to oracles that implement users algorithms with users key information. We consider two types of oracles.

Authentication oracles (A–oracles) implement the authentication algorithm with the signer's key. When presented with an *Authentication query (A–query)*, consisting of a source state $s \in \mathcal{S}$, the A–oracle generates the signed message $m = (s, t)$ (or just the signature t).

The impersonation and substitution attacks in traditional A–codes correspond to the case that 0 and 1 A–queries are allowed, respectively.

Verification oracles (V–oracles) implement the verification oracle with a particular verifier's key (as in SHZI02). On input (s, t), the V–oracle generates a TRUE/FALSE result. The queries to this oracle are called *V–queries.*

If the arbitration algorithm is different for the verifier's algorithm, we also need to consider an arbitration oracle.

In symmetric A–codes, A–oracles and V–oracles have the same information; i.e. they implement the same algorithm with the same keys but in asymmetric systems, the oracles have different keys.

A V–query against a verifier U_i gives information about the verification key of U_i. If verifiers use the same verification algorithm with different keys chosen using the same algorithm (for example random selection with uniform distribution), then the average information from a query will be the same for the two queried verifiers.

Attacks will be against a *target verifier*. The V–queries against this verifier will intuitively be expected to be more 'useful' than a query against a non–targeted verifier. Thus we define **Type V_1–queries (V_2–queries)**] as being made to a non–targeted (targeted) verifier.

Security Evaluation:

Let $e_C = \{e_j : j \in C\}$ be the colluders key set. $P_D^{t_{V_1},t_{V_2}}$, $P_{RS}^{t_A,t_{V_1},t_{V_2}}$ and $P_{RF}^{t_a,t_{V_1},t_{V_2}}$ denote success probabilities given t_A A–queries, t_{V_1} V_1–queries to each non–targeted verifier and t_{V_2} V_2–queries. Let $Q(t_A, t_{V_1}, t_{V_2})$ and $R(t_A, t_{V_1}, t_{V_2})$ denote the sequence of queries and responses, respectively and let $(Q,R)(t_A, t_{V_1}, t_{V_2})$ denote the pair of queries and responses.

$$P_D^{t_{V_1},t_{V_2}} = \max_{U_i} \max_{C \subset \mathcal{U}} \max_{\substack{e_C, m \\ m \notin M(e_t) \\ Q(t_{V_1}, t_{V_2})}} P(m \text{ is valid for } U_i, \text{ invalid for } U_A | e_C, (Q,R)(t_{V_1}, t_{V_2}))$$

$$P_{RS}^{t_A,t_{V_1},t_{V_2}} = \max_{\substack{C \subset \mathcal{U} \\ U_i \notin C}} \max_{\substack{e_C, m \\ Q(t_A, t_{V_1}, t_{V_2})}} P(m \text{ is valid for } U_i | e_C, (Q,R)(t_A, t_{V_1}, t_{V_2}))$$

$$P_{RF}^{t_A,t_{V_1},t_{V_2}} = \max_{\substack{C \subseteq \mathcal{U} \\ U_i}} \max_{\substack{e_C, m \\ Q(t_A, t_{V_1}, t_{V_2})}} P(m \text{ is valid for } U_0 | e_C, (Q,R)(t_A, t_{V_1}, t_{V_2}))$$

We say a system is $(\epsilon, w, n, t_A, t_{V_1}, t_{V_2})$*–secure* if the success chance of the best attack when t_A queries of type A, t_{V_1} queries of type V_1 and t_{V_2} queries of type V_2 are allowed, is at most ϵ.

Adaptive and non–adaptive queries

In the model we allow the queries to be asked in an arbitrary order. The success probability considers all possible interactions involving t A–queries, t_{V_1} V_1–queries and t_{V_2} V_2–queries and is maximised as the attacker's best strategy.

MRA^3**–codes** are similarly defined but the arbiter may in the collusion. In our model we assume the arbiter has the key information of a verifier. This means security of an MRA^3–code against a collusion containing U_A and w verifiers can be achieved by a $(\epsilon, w+1, n)$–MRA^2–code. Generally, success probability of the collusion attacks involving a dishonest arbiter must be considered.

A–queries and V–queries

Although distinguishing among the query type is important for efficiency of constructions, we can guarantee some security against V–queries even if we only consider A–queries. The following Lemma shows protection against V_1–queries can be obtained by constructing codes providing protection against larger collusions.

Lemma 1. *An* $(\epsilon, w, n, t, 0, 0)$*–*$MRA^2$ *provides* ϵ*–security against collusions of size* $w - v$*, assuming colluders can have* t *A–queries and any number of* V_1*–queries against* v *verifiers.*

This result follows since the information gained by V_1–queries to U_i at most equals the key held by U_i, which would be yielded up were U_i in the collusion.

V_2–queries provide information on the target verifier's key. For secure codes, one expects to obtain less information from queries resulting in FALSE compared to those giving TRUE. This is since the probability of the former type of queries is expected to be higher than that of the latter.

3.2 Security notions in HSZI00 and SHZI02

One main aim of developing our framework is to unify USDS, including SHZI02. We address this here. HSZI00 correctly recognised the inadequacy of MRA and $DMRA$–codes as USDS and argued that multireceiver A–codes make sense only in a broadcast environment [7, p.132] and [8, p.69].

The term 'multireceiver' in the A–code context refers to the property: *any receiver who receives the authenticated message can verify it.* This is exactly as required in signature schemes. Multireceiver schemes do not 'require' that the signed message be received simultaneously by all group members. Rather they guarantee that *if* any group member receives the signed message then they can verify it. However, as noted earlier, MRA–systems assume a trusted sender and so do not provide security against attacks by collusions including a distrusted signer. The model proposed in section 3.1 assumes the signer is distrusted.

The following Lemma shows we need not consider 'transfer with a trap' (so named by HSZI00) attack. In a 'transfer with a trap' colluders construct a forged message that is acceptable to U_i and not U_j or U_A, and so when U_i presents the message to U_j, U_i is trapped. Here the colluders may include the signer.

Lemma 2. *The success probability in 'transfer with a trap' is at most equal to* $\max\{P_D^{t_{V_1},t_{V_2}}, P_{RS}^{t_A,t_{V_1},t_{V_2}}\}$.

Proof. If the signer is part of the collusion the attack succeeds if *(i)* the message satisfies the requirement for a successful denial attack, and *(ii)* is furthermore unacceptable to some receiver U_j. If the signer is not part of the collusion the attack succeeds if *(i)* the message satisfies the requirement for a successful spoofing attack, and *(ii)* is not acceptable to both the receiver U_j and the arbiter U_A. Success in transfer with a trap requires two conditions to be satisfied and thus has less chance of success than plain denial or spoofing attacks, respectively.

SHZI02 introduced a wide range of new security notions closely following computational models. They considered the 'strongest security notion' for their proposed construction. In our model of asymmetric group A–codes, we consider the most powerful collusion, with the most useful information, using their best strategy, with success defined by success against a single verifier. The most powerful collusion includes the signer and the arbiter, with their key information and access to oracle queries, and the attack goal is constructing 'a message' acceptable to 'a verifier' (in SHZI02 notation, existential forgery and existential acceptance). This is the same as the 'strongest security notion' in SHZI02.

Other types of forgeries in SHZI02 are *Total break* and *selective forgery* which are harder to achieve and, while expressible in our framework, are of less interest. Similarly, colluders information can be restricted to key information only (*Key–only attacks*); i.e. disallow queries. As mentioned earlier, we consider all valid query sequences (§3.1), so adaptive queries need not be considered.

SHZI02 define other security goals (*Total* and *selective* acceptance), both harder to achieve than the *existential acceptance* considered in our model and used in the 'strongest security notion'.

SHZI02 [15] note "*the strongest signature scheme is one secure against existential acceptance forgery under adaptive chosen message attack and adaptive chosen signature attacks*", and use this model for their constructions. The security model of MRA^3–codes, matches this definition. In section 5 we give a language to express a wide range of security models in authentication scenarios. The value of particular scenarios depends on practical applications.

Information theoretic bounds

Establishing the relationship between USDS in HSZI00 and SHZI02 models and multireceiver codes allows us to derive information theoretic bounds for USDS. We give bounds for the attacks defined in section 3.1. Since the arbiter is treated as having a verifier's information, the bounds for arbiter inclusive attacks are the same as the bounds for a collusion of size $w+1$. These bounds consider A–queries only and so the query set is $Q(t_A)$, with $(Q,R)(t_A)$ the message and response set. We use $M' = M \backslash Q(t_A)$ to denote the rest of the message space and E_C for the keyspace of colluders.

$$P_D \geq 2^{-I(M;E_i,E_A|E_C)} \qquad P_{RS}^{t_A} \geq 2^{-I(M';E_i|E_C,(Q,R)(t_A))}$$
$$P_{RF}^{t_A} \geq 2^{-I(M';E_T|E_C,(Q,R)(t_A))}$$

The bounds when V–queries are considered remains an open problem.

$$\begin{tikzcd}
 & & & (\epsilon,w,n,t_A,t_{V_1},t_{V_2})\text{–secure} \arrow[dl] \\
(\epsilon,w,n,t_A)\text{–}MRA & (\epsilon,w,n,t_A)\text{–}MRA^2 \arrow[l] & (\epsilon,w,n,t_A)\text{–}MRA^3 \arrow[l] & \\
 & & & (\epsilon,w,n,t_A,t_V)\text{–secure} \arrow[ul]
\end{tikzcd}$$

Fig. 1. The relationship between different types of security notions for authentication codes. We use $A \rightarrow B$ to imply that a code of type A satisfies the security requirements of a code of type B. All codes, except for (ϵ,w,n,t_A)–MRA, are types of USDS. The (ϵ,w,n,t_A,t_V) code satisfies the strongest security notions of SHZI02 with t_A A–queries, t_V V_1 queries and $t_V - 1$ V_2 queries. We note the two rightmost USDS are essentially the same and the distinction lies in seperating V_1 and V_2–queries.

4 Constructions

In constructing group A–codes the challenge is to have *secure and efficient* constructions. Optimal constructions meet minimum requirements for keys and have the shortest signature length, but are rare and inflexible. ϵ–security gives guaranteed security without the highest efficiency, but with the advantage of providing flexibility and a wide range of constructions.

Proof of security for systems with complex security goals is generally difficult. We give two algebraic methods of constructing group–based A–codes from simpler A–codes. The constructions use *polynomial codes* where signature generation and verification can be expressed by evaluation of multivariate polynomials over a finite field F_q with q elements. We assume all polynomials are in $F_q[x_1, \cdots x_n]$, the ring of polynomials over the finite field F_q. Constructions $\mathbf{C_0}$ and $\mathbf{C_1}$ are polynomial codes. Polynomial codes are generally efficient and often optimal.

A polynomial code can be expressed in terms of polynomials generated by the trusted authority (TA) during the **Key generation (KeyGen)** phase. The signer receives a signing polynomial $A(x, z)$ for generating signatures. Each receiver U_i gets a verification polynomial and some identification information u_i. The identifier may be public (private) if the sender is trusted (distrusted).
Signature generation (SigGen): The signature of a source s is $\alpha(z) = A(s, z)$. We assume authentication codes without secrecy so the signed message is $(s, \alpha(z))$.
Signature verification (SigVer): A receiver U_i accepts a signed message iff $\alpha(z)|_{z=u_i} = V_i(x)|_{x=s}$.

4.1 A systematic approach to constructing group A–codes

We use multiple instances of a component code, combined using powers of a single variable, or using distinct variables for each instance. We consider two synthesis algorithms, Σ_1 and Σ_2.

Synthesis algorithm: $\boldsymbol{\Sigma_1}$

KeyGen: The TA generates $k+1$ instances of the component authentication code. For each instance j, a component signing key $A_j(x, z)$ and component verification keys, $V_{ij}(x)$ for each verifier i are generated, such that $V_{ij}(x) = A_j(x, u_i)$ where $u_i \in \mathbb{F}_q$ is U_i's identifier. The TA gives U_0 the polynomial

$$B(x, z, y) = \sum_{j=0}^{k} A_j(x, z) y^j$$

and each verifier U_i another identifier u'_i, if necessary, and a polynomial

$$W_i(x) = \sum_{j=0}^{k} V_{ij}(x)(u'_i)^j = B(x, u_i, u'_i) \ .$$

SigGen: The signature of a source state s is $\alpha(z,y) = B(s,z,y)$.
SigVer: A receiver U_i accepts a signed message iff $\alpha(z,y)|_{z=u_i,y=u_i'} = W_i(x)|_{x=s}$.

Discussion and example for Σ_1

Σ_1 can be used to construct codes that provide protection for multiple receivers, construct asymmetric codes from symmetric codes, and construct dynamic sender codes from single sender codes.

We shall consider synthesis of an MRA–code from a two party A–code. The approach also be used to construct HSZI00 (dynamic sender) from a (ϵ, w, n, t_A)–secure code providing protection against collusions of size w and t A–queries.

Let the component code be $\mathbf{C_0}$, where the signer has $A(x) = a + bx$ and $V(x) = A(x)$. Using Σ_1 we obtain an authentication code as follows.
KeyGen: The TA generates $k+1$ instances of the code $\mathbf{C_0}$, specified by $A_j(x) = a_j + xb_j, 0 \le j \le k$. The TA gives U_0 the polynomial

$$B(x,y) = \sum_{j=0}^{k} (a_j + b_j x) y^j$$

and each verifier U_i an identifier u_i and verification polynomial

$$W_i(x) = \sum_{j=0}^{k} (a_i + b_i x)(u_i)^j \ .$$

SigGen: The signature for a source state s is $\alpha(y) = B(s,y)$.
SigVer: User U_i accepts $(s, \alpha(y))$ iff $\alpha|_{y=u_i} = W_i(x)|_{x=s}$.

The above construction is the same as the $(\epsilon = 1/q, k, n)$–secure MRA–code of [4]. This follows since the signature generation function can be written as $B_i(x,y) = \sum_j a_j y^j + x \sum_i b_j y^j = f(y) + xg(y)$. If u_i is only known to the receiver, we have an (ϵ, k, n)–MRA^2–code, with $\epsilon = 1/(q-k)$, since the signer cannot deny a signature.

Synthesis algorithm: Σ_2

KeyGen: The TA generates $k+1$ instances of the component authentication code. For instance j, a signing key $A_j(x,z)$ and verification keys, $V_{ij}(x) = A_j(x, u_i)$, for each verifier i, are generated. The TA gives U_0 the polynomial

$$B(x,z,\mathbf{Y}) = \sum_{j=0}^{k} A_j(x,z)\mathbf{Y}_j$$

and each verifier U_i an identifier u_i, randomly generated vector $\mathbf{v_i} \in \mathbb{F}_q^{k+1}$ also written as $\mathbf{v_i} = (v_{i0}, v_{i1}, \ldots, v_{ik})$, and a verification polynomial

$$W_i(x) = \sum_{j=0}^{k} V_{ij}(x) v_{ij} \ .$$

SigGen: The signature of a source state s is $\alpha(z, \mathbf{Y}) = B(s, z, \mathbf{Y})$.
SigVer: A receiver U_i accepts a signed message iff $\alpha|_{(z=u_i,\mathbf{Y}=\mathbf{v_i})} = W_i(x)|_{x=s}$.

Discussion and example for Σ_2

This algorithm allows one to construct asymmetric codes from symmetric ones, multireceiver codes from single receiver codes, or dynamic codes from single sender codes. Again we consider constructing an MRA–code from a two party A–code. As before we use $\mathbf{C_0}$ as the component code.
KeyGen: The TA randomly generates $k+1$ instances of the code $\mathbf{C_0}$, specified by the polynomial $A_j(x) = a_j + xb_j, 0 \le j \le k$. The TA gives U_0 the polynomial

$$B(x, \mathbf{Y}) = \sum_{j=0}^{k} A_j(x)\mathbf{Y}_j$$

and each U_i an identifier $u_i \in \mathbb{F}_q$, a randomly generated vector $\mathbf{v_i} \in \mathbb{F}_q^{k+1}$, and a polynomial

$$W_i(x) = \sum_{j=0}^{k} V_{ij}(x)v_{ij} \ .$$

SigGen: The signature for a source state s is $\alpha(\mathbf{Y}) = B(s, \mathbf{Y})$.
SigVer: User U_i accepts $(s, \alpha(\mathbf{Y}))$ iff $\alpha(\mathbf{Y})|_{\mathbf{Y}=\mathbf{v_i}} = W_i(s)$.

Theorem 1. *The above construction is an (ϵ, w, n)–MRA–code. The authenticator and key sizes for signer and user are $k+1, 2(k+1)$ and $k+3$ respectively. In this case $\epsilon = 1/q$.*

Intuitively this result follows since each copy of the two party code provides security for a single colluder and for each colluder one copy of the code is added. Compared to $\mathbf{C_1}$, obtained using Σ_1, this construction has a larger key size for verifiers but the same signer key size and the same signature length.

Σ_2 construction can also be used to provide protection against V–queries. This property will be used in synthesising SHZI02 (§4.3). To show this property we re–visit the construction above and show it can be seen as an $(\epsilon, 0, n, 1, t_{V_1} = k+1, t_{V_2} = k)$–secure code. That is, a code where signer is distrusted but verifiers are trusted. This is dual to traditional MRA–codes where the signer is trusted and verifiers collude. The most powerful attack is the signer's denial attack against a verifier. The signer does not know the identity vector $\mathbf{v_i}$ and has to construct a pair $(s, \alpha'(\mathbf{Y}))$ such that (i) $\alpha'(\mathbf{v_j}) = W_j(s')$ and (ii) $\alpha(\mathbf{v_j}) \neq B(s, \mathbf{Y})$. He can have k V_2–queries. The V_1 queries give information about the key information of other verifiers only. The signer attempts to construct a message $(s, \alpha'(\mathbf{Y}))$ such that (i) $\alpha'(\mathbf{v_j}) = W_j(s')$ and (ii) $\alpha(\mathbf{v_j}) \neq B(s, \mathbf{Y})$.

Each V_2–query gives a tag $\alpha_i\mathbf{Y}, 0 \le i \le k-1$ such that $\alpha_i(\mathbf{v_j}) \neq W_j(s')$, i.e. a source state, tag pair unacceptable to U_j. The adversary can choose k α_i so $\alpha_i(\mathbf{v_j}) = \alpha_l(\mathbf{v_j})$ if and only if $i = l$, so each tag tests a different value against

$W_j(s')$. Each of the tags used reduces the possible values of $W_j(s')$ by 1. Thus the probability of the adversary choosing a tag acceptable to U_j is $\epsilon = 1/(q-k)$.

This shows one may apply Σ_2 to $\mathbf{C_0}$ to obtain either a $(\epsilon, k, n, 1, 0, 0)$–secure or a $(\epsilon, 0, n, 1, k+1, k)$–secure code. Indeed, though we shall not give details here, the Σ_2 synthesis gives an $(\epsilon, k_1, n, 1, k_2+1, k_2)$–secure code, where $k_1 + k_2 = k$.

4.2 Construction of USDS

Σ_2 can be applied to the A^2–code and A^3–codes in [9] to construct MRA^2 and MRA^3–codes from $\mathbf{C_1}$. We omit the details and instead show how to use a synthesis approach similar to Σ_1 on source states rather than on identities to synthesise MRA^2 and MRA^3–codes that protect against higher number of queries. That is we show how to construct a $(\epsilon, w, n, t_A, 0, 0)$–secure code from a $(\epsilon, w, n, 1, 0, 0)$–secure code. A similar argument applies to MRA^3–codes when the arbiter has the key information of a verifier.

Theorem 2. *The construction* $\mathbf{C_1}$ *is an* (ϵ, w, n)*–secure* MRA^2*–code if* u_i *are known only to* U_i*. We have* $\epsilon = w/(q-w)$.

We call this construction $\mathbf{C_1^2}$. The security proof uses the knowledge that the strongest collusion consists of the signer and w verifiers whose aim is to construct an authenticator $\alpha(x)$ (a polynomial of degree w) such that $\alpha(u_j) = f(u_j) + sg(u_j)$ for some j. The result follows since while colluders know $f(x)$ and $g(x)$ they cannot determine the identity u_j of U_j. The construction guarantees ϵ–security if for given security ϵ and w we have $q \geq w(1 + 1/\epsilon)$. To construct an (ϵ, w, n, t_A)–secure MRA^2–code we use $t_A + 1$ copies of $\mathbf{C_1^2}$ and apply a modified version of Σ_1. (Similarly for MRA^3 from $\mathbf{C_1^3}$.)
KeyGen: The TA generates $t+1$ independent $\mathbf{C_1^3}$, $f_i(x) + zg_i(x)$, and gives U_0

$$B(x, y, z) = \sum_{k=0}^{t} (f_k(x) + zg_k(x))y^k = \sum_{k=0}^{t} \sum_{i=0}^{w} \sum_{j=0}^{1} a_{kij} x^i z^j y^k \ .$$

The TA gives verifier U_i a private $u_i \in F_q$ and $B(u_i, y, z)$. The arbiter has the key information of a verifier, that is $B(u_a, y, z)$ where u_a is the arbiters identifier.
SigGen: The signature of a source state $s \in \mathbb{F}_q$ is $\alpha(x, z) = B(x, s, z)$.
SigVer: User U_i accepts the message as authentic iff $\alpha|_{x=u_i} = B(u_i, y, z)|_{y=s}, \forall z$.

The key sizes for the signer and each verifier are $2(t+1)(w+1)$ and $2t+3$, respectively. The tag length is $2(w+1)$. As before appropriate choices of parameters can provide ϵ–security for any chosen ϵ.

Theorem 3. *The above construction is a* MRA^3*–codes that protects against* t *A–queries with* $\epsilon = w/(q-w)$.

This code is similar to a generalised $\mathbf{C_1}$ construction given in [13, §5.1] as an MRA–code protecting against multiple A–queries.

4.3 USDS constructions: The SHZI02 model

SHZI02 gave a construction that satisfies their proposed 'strongest security notion'. We construct a code with the same security level using the synthesis methodology above. The main advantage of this description is that the security proof can be straightforwardly derived from that of the underlying codes.

The SHZI02 model uses the same setting as MRA^3–codes. For an attack against U_j, by a collusion of w out of n verifiers, the adversary may have (i) t A–queries, (ii) t' V_1 queries from each verifier other than U_j, and (iii) $t'-1$ V_2–queries rejected by U_j.

The synthesis has two steps: (i) constructing an $(\epsilon, 0, 2, t, 0, 0)$–secure code, and (ii) constructing a code with $(\epsilon, w, n, t, t', t'-1)$–security.

We start from $\mathbf{C_0}$: a component code that is $(\epsilon, 0, 2, 1, 0, 0)$–secure. The key is a pair of random numbers $(a, b) \in \mathbb{F}_q^2$ shared by the signer and verifier. Using the synthesis akin to Σ_1, described in the previous section, we take $t+1$ copies, thus $A_i(x) = a_i + b_i x$, we obtain an $(\epsilon, 0, 2, t, 0, 0)$–secure code, where the polynomial held by the signer and by each verifier (noting they are still all trusted), is

$$B(x, y) = \sum_{i=0}^{t} A_i(x) y^i \ = f(y) + x g(y)$$

where $f(y) = \sum_{i=0}^{t} a_i y^i$ and $g(y) = \sum_{i=0}^{t} b_i y^i$.

The signature for a source state s is $\alpha(x) = B(x, s)$, and a message is accepted if $\alpha(x) = B(x, s)$. Let this $(\epsilon, 0, 2, t, 0, 0)$–secure code be the component code, and apply Σ_2 to $t'+w+1$ copies $B_i(x, y), 0 \le i \le t'+w$. The TA gives U_0

$$C(x, y, \mathbf{Y}) = \sum_{j=0}^{w+t'} B_j(x, y) \mathbf{Y}_j$$

and verifier U_i a randomly chosen identity $\mathbf{v_i} \in \mathbb{F}_q^{w+t'+1}$ and verifying polynomial

$$W_i(x, y) = C(x, y, \mathbf{v_i}) = \sum_{j=0}^{w+t'} B_j(x, y) \mathbf{v_i}_j \ .$$

SigGen: The signature for a source state s is $\alpha(x, \mathbf{Y}) = B(x, s, \mathbf{Y})$.
SigVer: User U_i accepts $(s, \alpha(x, \mathbf{Y}))$ iff $\alpha(x, \mathbf{Y})|_{\mathbf{Y}=\mathbf{v_i}} = W_i(x, y)|_{y=s}, \forall x$.

We may write the complete key of the signer as

$$C(x, y, \mathbf{Y}) = \sum_{i=0}^{t} \sum_{j=0}^{w+t'} \sum_{k=0}^{1} A_{ijk} \mathbf{Y}_j y^i x^k \ .$$

This *is* the construction of SHZI02, satisfying the 'strong security notion' and constructed using Σ_1 and Σ_2. We used Σ_1 to synthesise a t–message system from a 1–message code. We used Σ_2 to synthesise an asymmetric system secure against collusions of up to size w, and t' V–queries. Collusions may include the signer, or arbiter in our model, and the arbiter has a verifier's key.

SHZI02 note this code meets the $1/q$ bound on security, although it is not known to be optimal. Rather than starting with $\mathbf{C_0}$ we could omit the Σ_1 step and use an optimal $(\epsilon, 0, 2, t, 0, 0)$–secure code, with signer polynomial $B(x) = \sum_{i=0}^{t} A_i y^i$ [9]. We omit details but synthesising this code using Σ_2 as above gives a $(1/(q-t'), w, n, t, t', t'-1)$–secure code. The authenticator, signer's and verifier's keys sizes are, $(w+t'+1)$, $(t+1)(w+t'+1)$ and $(w+t'+1)+(t+1)$, respectively, half those of the SHZI02 as formulated above. While information theoretic and combinatorial bounds are not yet known for these codes, it seems unlikely the construction of SHZI02, as developed above, is optimal.

5 Generalised authentication codes

A general setting for Generalised A–codes (GA–codes) consists of a set $\mathcal{U}$ of participants, each with some secret key information, such that any group member may sign a message and verify signed messages. To emphasise the new aspects of GA–codes, we assume there is one signer, the approach can be extended to dynamic signer systems. The set $\mathcal{U}$ contains n verifiers, an arbiter U_A, and the signer U_0. Let E_X denote the set of all possible keys values held by a set X of participants. We use $\mathcal{M}(E)$ to denote the set of messages valid under all the keys in E. An adversary corrupts some subset of participants that will form a *colluding set*. We assume these sets are staticly determined.

We consider codes without secrecy, where the authenticated message for a source state s can be written in the form (s, t), where t is a tag or authenticator.

Generalised oracles: We generalise the oracles of section 3.1 by defining *generalised A–oracles* and *generalised V–oracles* that can generate and verify, respectively, messages of defined *type.*

Message type: We say a message m is of type $\tau = (e_{i_1}, \ldots, e_{i_l}; e_{j_1}, \ldots, e_{j_{l'}})$, if

$$m \in \{\widehat{\mathcal{M}}(e_{i_1}) \cap \ldots \cap \widehat{\mathcal{M}}(e_{i_l})\} \backslash \{\widehat{\mathcal{M}}(e_{j_1}) \cap \ldots \cap \widehat{\mathcal{M}}(e_{j_{l'}})\} \tag{1}$$

where $\widehat{\mathcal{M}}(e) \subseteq \mathcal{M}(e)$. In other words m is valid for $(e_{i_1}, \ldots, e_{i_l})$ and not valid for $(e_{j_1}, \ldots, e_{j_{l'}})$. This captures exclusions of already 'used' queries from the message space. A message type is NULL if the types message space is empty.

A gA–oracle takes a source state and type τ, and generates an authenticated message (source state followed by the signature) of type τ, or outputs NULL, if it is not possible to generate such a message. A gV–oracle takes a message m and a type τ and produces a TRUE result if m is of type τ, and FALSE otherwise.

Since the status of a message with respect to the arbiter is also relevant, one may have messages known to be acceptable or unacceptable to the arbiter by considering inclusion in $\mathcal{M}(e_A)$. If the arbitration algorithm differs from the verification algorithm, arbiter queries need to be considered separately.

Collusion structure: The collusion structure is written as a pair (C, Φ_C), where C is a colluding set and Φ_C determines the oracle queries accessible to C. The set Φ_C contains a list of message types ϕ_i, multiplicities ℓ_i and a flag ρ_i that determines if the query is made to the gA–oracle or to the gV–oracle. For each i, ℓ_i messages of type ϕ_i may be queried to an oracle of type ρ_i. Let $\mathcal{R}(\phi_i)$ be the set of input and response pairs associated with the ϕ_i queries.

A $(\epsilon, w, n, t_A, t_V)$*–threshold* collusion structure is a collusion structure in which a colluding set contains at most w verifiers and has access to up to t_A A–queries, up to $t_V - 1$ V_2–queries (from the targeted verifier) and up to t_V V_1–queries (from each other verifier). A collusion set may also include the signer, and/or the arbiter.

The **Goal of an attack** is specified by the type of message to be constructed by the colluders.
An **Authentication Scenario** $\sigma(\mathcal{C})$ is defined by a set of participants, a collusion structure, and the protection the system can provide against colluder's attacks. Performance of an authentication scenario against a colluding set C with goal type γ is measured by $P(\gamma|\Phi_C)$, the highest success chance of a collusion with message set Φ_C. The success probability of such an attack is defined as

$$P(\gamma; \Phi_C) = \max_{\mathcal{C}} \max_{e_C \in E_C} \max_{\phi_C \in \Phi_C} P(m \text{ is of type } \gamma | \mathcal{R}(\phi_C), e_C)$$

where $P(m(\gamma)|\mathcal{R}(\phi_C), e_C)$ is the probability of generating the message m of type γ given queries with responses, $\mathcal{R}(\phi_C) \in \mathcal{R}(\Phi_C)$, specified by the collusion structure ϕ_C, and key information $e_C \in E_C$. We note that for gA–queries only the space $\mathcal{R}$ reduces to the message space $\mathcal{M}(\Phi_C)$, as below.

Information theoretic bounds

The attack probability bounds for $A, A^2, A^3, MRA, MRA^2, tA^3$ and $tMRA^2$ codes, at least may be concisely represented using authentication scenarios;

$$P(\gamma; \Phi_C) \geq 2^{-I(M(E_\gamma); E_\gamma | M(\Phi_C), E_C)} .$$

6 Concluding remarks

We proposed an extension of traditional A–codes and showed the resulting framework encompesses the recently proposed USDS schemes, and all the previously known ones, hence unifying all models and constructions in the area. Introducing the notion of V–queries suggests an interesting model for attacker's strategy in A–codes not previously considered. This is hence a rich area for research.

We also developed an algebraic method for synthesizing group A–codes from simpler component codes, which removes the shortcoming of previous synthesis constructions. We gave two general methods, called Σ_1 and Σ_2, and gave an example construction using each.

We believe our work fills a gap in understanding USDS and provides a unified framework for USDS and their future extensions.

Acknowledgements

This work is supported in part by an Australian Research Council Discovery Grant (ARC Ref No. A10012057).

References

1. E. F. Brickell and D. R. Stinson 'Authentication codes with multiple arbiters.' *Eurocrypt'88* LNCS **330**, (Springer–Verlag, 1988) 51–5.
2. D. Chaum and S. Roijakkers 'Unconditionally–secure digital signatures.' *Crypto'90* LNCS **537** (Springer, 1990) 206–15.
3. Y. Desmedt and M. Yung 'Arbitrated unconditionally secure authentication can be unconditionally protected against arbiter's attack.' *Crypto'90* LNCS **537** (Springer, 1990) 177–88.
4. Y. Desmedt, Y. Frankel and M. Yung 'Multi–receiver/multi–sender network security: efficient authenticated multicast/feedback.' *IEEE Infocom'92* (1992) 2045–54.
5. H. Fujii, W. Kachen and K. Kurosawa 'Combinatorial bounds and design of broadcast authentication.' *IEICE Trans.* **E79–A**(4) (1996) 502–6.
6. E. N. Gilbert, F. J. MacWilliams and N. J. A. Sloane 'Codes which detect deception.' *Bell System Tech. J.* **53**(3) (1974) 405–24.
7. G. Hanaoka, J. Shikata, Y. Zheng and H. Imai 'Unconditionally secure digital signature schemes admitting transferability.' *Asiacrypt'00* LNCS **1976**, (2000) 130–42.
8. G. Hanaoka, J. Shikata, Y. Zheng and H. Imai 'Efficient and unconditionally secure digital signatures and a security analysis of a multireceiver authentication code.' *PKC'02* LNCS **2274**, (2002) 64–79.
9. T. Johansson 'Contributions to unconditionally secure authentication.' Ph.D. Thesis, (Lund University, Sweden, 1994).
10. T. Johansson 'Lower bounds on the probability of deception in authentication with arbitration.' *IEEE Trans. Inform. Theory.* **40**(5) (1994) 1573–85.
11. T. Johansson 'Further results on asymmetric authentication schemes' *Information and Computation* **151** (1999) 100–33.
12. J. Rompel 'One–way functions are necessary and sufficient for secure signatures.' *STOC'90* (1990) 387–94.
13. R. Safavi–Naini and H. Wang 'Multireceiver authentication codes: Models, bounds, constructions and extensions.' *Information and Computation* **151** (1999) 148–72.
14. R. Safavi–Naini and H. Wang 'Broadcast authentication for group communication.' *Theoretical Computer Science* **269** (2001) 1–21.
15. J. Shikata, G. Hanaoka, Y. Zheng and H. Imai 'Security Notions for Unconditionally Secure Signature Schemes' *Eurocrypt'02* LNCS **2332** (2002) 434–49.
16. G. J. Simmons 'Authentication theory/coding theory' *Crypto'84* LNCS **196** (Springer–Verlag, 1984) 411–31.
17. G. J. Simmons 'A Cartesian product construction for unconditionally secure authentication codes that permit arbitration' *J. Crypt.* **2**(2) (1990) 77–104.
18. B. Smeets 'Bounds on the probability of deception in multiple authentication.' *IEEE Trans. Inform. Theory* **40**(5) (1994) 1586–91.
19. Y. Wang and R. Safavi–Naini 'A^3–codes under collusion attacks' *Asiacrypt'99*, LNCS **1716** (Springer, 1999) 360–98.