Distributed Delegation Systems

Chris Charnes
Josef Pieprzyk *
Rei Safavi-Naini

Centre for Computer Security Research
Department of Computer Science
University of Wollongong
Wollongong, NSW 2500, AUSTRALIA

e-mail: charnes/josef/rei@cs.uow.edu.au

November 6, 1995

Abstract

In this paper we introduce unconditionally secure distributed delegation systems that allow groups of participants to collectively authorize a delegate to sign a message (construct an authentic message) on their behalf. We propose a model and investigate its properties when delegation is valid for one transmission. The model is extended to multiple delegation and constructions with more economical key exchange are proposed.

1 Introduction

In many situations it is required to authorize a principal to perform a specific action for a specified period of time. For example, a director may want to authorize a deputy to sign a document on his behalf in an emergency. In a computer environment users routinely need to delegate applications to read, write to, or execute on their behalf. Delegation is discussed in more detail in [1].

In [5] Desmedt introduced the concept of group oriented cryptography where an active participant who does cryptographic operations is replaced by a group of participants who collaborate to perform the same cryptographic operation. Group oriented cryptography is particularly useful in the context of distributed computer systems where frequently an action, for example signing a message [3] or encrypting a cryptogram [8], requires the collaboration of a number of participants.

There are two main approaches in modeling and assessment of group-oriented systems: unconditional security where there is no limitation on the computational power of the enemy, and computational security, where security relies of excessive effort required by an enemy who wants to breach the security of the system. While the first approach does not usually produce practically viable constructions but does allow a better understanding of security parameters of the system and in most cases could be used to construct secure systems. For example study

*Support for this project was provided in part by the Australian Research Council under the reference number A49530480

of unconditionally secure authentication codes, and their relation with universal classes of hash functions, has resulted in message authentication codes that are not only efficient and practical for arbitrary length messages, but have also proven security properties [2]. We will consider unconditionally secure systems in this paper.

In this paper we introduce distributed delegation system (DD-systems) in which a group of participants authorizes a delegate to construct an authentic message on behalf of the group. In its basic form the delegation is valid only for a single transmission and the delegate needs re-instantiation of its authorization before the construction of the next authentic message. DD-systems differ from distributed authentication systems in that the content of the authentic message is determined by the delegate instead of the transmitting participants. Once the authority is transferred to the delegate, it can choose any message it wants and authenticate it, while in a distributed authentication system [6] the combiner who constructs the authentic message, needs to obtain shadows of the message to be authenticated from participants belonging to an access set. In other words, in DD-systems the delegate takes the role of the valid transmitter for one message, but in distributed authentication systems the combiner is always dependent on inputs from authorized participants. In contrast to distributed authentication systems, DD-systems are simple and easier to design and implement. The reliability and security of both DD-systems and distributed authentication systems rest on a trusted party so from this point of view they are equivalent.

In [4] Croft and Harris proposed a distributed authorization system in which a group of t out n signatories authorize a transaction to proceed. Their system uses Shamir's secret sharing scheme [10] and the one-way property of exponentiation to issue shadows to the participants which are used for a single transmission. This system can be viewed as a distributed delegation system whose security is conditional; it depends on the computational difficulty of the discrete logarithm.

In this paper we consider unconditionally secure DD-systems which can be used for single and multiple transmissions.

2 Background

Assume that there are two parties: a transmitter and a receiver who want to establish an authentication channel between themselves. In the model of authentication proposed by Simmons [7], an authentication matrix is used to construct authentic codewords for source states. The communicants share a secret key which allows the transmitter to construct an authentic codeword and the receiver to verify the authenticity of the received codeword. Secrecy of the key protects the system against the spoofing attack. We consider authentication without secrecy and assume that codewords corresponding to source states are constructed by appending suitable tags to the source states.

In [5] Desmedt introduced the idea of distributed authentication where the transmitter is replaced by a group of transmitters. A subset of transmitters send their partial codewords to a combiner who constructs the codeword and communicates it to the receiver. The combiner can always compute the correct codewords if the subset of participants is authorized (belongs to the access structure). The calculations fail with high probability whenever the subset of participants does not belong to the access structure.

In this paper we consider *distributed delegation systems*. In a delegation system with a single transmitter, the delegate may act on behalf of the transmitter, performing certain operations for a specified period of time. We will consider the problem of delegating the authority to sign or authenticate an arbitrary message for a single transmission. In distributed delegation, the

delegate has to be authorized by a group of participants (transmitters) to construct a single authentic codeword for a message on their behalf. The receiver is able to verify the authenticity of the codeword based on a public authentication matrix and knowledge of a secret key.

A distributed delegation system consists of a trusted dealer (DE) and a delegate (D). The trusted dealer DE, randomly generates a secret key corresponding to a row of the public authentication matrix and sends the key to the receiver via a secure channel. DE also computes a collection of shares which are distributed to the participants via secure channels. Any authorized subset of participants can recalculate the secret key (secret encoding rule).

A subset of participants who want to authorize the delegate to authenticate a single source state proceed as follows. They send their shares (and possibly a source state) to the delegate via secure channel. The delegate collects shares from participants and computes the secret key. Knowing the encoding rule D can construct an authentic codeword, of the form $(source, state, tag)$, for any source state that it chooses and sends it to the receiver via insecure channel. A delegate can be a tamper-proof device which produces a codeword for a message and shares.

The receiver verifies the codeword by checking the received tag with the tag obtained from the authentication matrix and accepts the codeword if the two tags are the same. This protocol is valid for a single transmission, and must be repeated for subsequent transmissions.

In this paper we introduce a model for distributed delegation systems and describe constructions using secret sharing systems. In Section 5 we extend the model to include multiple delegation and propose a construction based on families of secret sharing systems.

The organization of the paper is as follows. Section 3 gives a model of distributed delegation systems. Properties of these systems together with solutions are discussed in Section 4. Section 5 defines ideal distributed delegation systems and their applications for multiple delegation. Conclusions are given in Section 6.

3 The model

The information source is a collection of k source states and is denoted by $\mathcal{S}$. The codewords sent over the channel are from a set $\mathcal{M}$, $|\mathcal{M}| = M$, and are constructed using a collection of encoding rules $\mathcal{E}$, $|\mathcal{E}| = E$.

Definition 3.1 *An encoding rule (also called a key) is a one-to-one mapping $e : \mathcal{S} \rightarrow \mathcal{M}$ that takes a source state s and generates a codeword $m = s \parallel t$, where $\parallel$ denotes concatenation and t is a* tag *selected from a set $\mathcal{T}$, $|\mathcal{T}| = T$.*

The encoding rule can be rewritten as $e : \mathcal{S} \rightarrow \mathcal{S} \times \mathcal{T}$ and the codeword $e(s) = s \parallel t$ for a source state $s \in \mathcal{S}$.

Definition 3.2 *An authentication matrix A is a $E \times T$ matrix whose rows and columns are indexed by encoding rules $e_1, \ldots, e_E$ and source states respectively. The entry $A(e, s)$ in the $e-th$ row and $s-th$ column of A is t if $e(s) = s \parallel t$. The codeword $m = e(s)$ is the valid codeword for the source state s under the encoding rule e.*

For the set of participants $\mathcal{P} = \{P_1, \ldots, P_n\}$, the family of all subsets $2^{\mathcal{P}}$ splits into two classes; the authorized subsets and the unauthorized subsets. A collection Γ of subsets of $\mathcal{P}$ is called the *access structure* and generates the class of authorized subsets as follows. We say that the set of participants $\gamma = \{P_{i_i}, \cdots P_{i_\ell}\}$ belongs to $\Gamma = \{\Gamma_1, \ldots, \Gamma_\alpha\}$ if there is a set $\Gamma_i \subseteq \gamma$ for some $i = 1, \ldots, \alpha$.

An access structure Γ can always be recreated from a collection of *minimal* subsets of participants. A minimal subset is an authorized subset which becomes unauthorized after removing a single participant. So we can write

$$\Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_u\}$$

where Γ_j for $j = 1, \ldots, u$, $|\Gamma_j| = \nu_j$, are the minimal subsets. This is called the *minimal representation* of Γ.

The participants know their shares of the secret and once a subset $\Gamma_v \in \Gamma$ combine their shares, they can recreate the secret – in this case the encoding rule (or the key).

Definition 3.3 *A Distributed Delegation system (DD system) consists of a dealer, a delegate and a group of n participants endowed with an access structure Γ with the following properties (Γ has the minimal representation).*

1. *The Dealer (DE) is trusted. The dealer selects randomly, with uniform probability distribution, a key (an encoding rule) from the set $\mathcal{E}$ of encoding rules and communicates it to the receiver via a secure channel. DE then computes shares of the key using the function*

   $$f_{DE} : \mathcal{E} \times \mathcal{P} \rightarrow \mathcal{C}$$

   which for a given key $e_j \in \mathcal{E}$, assigns shares $c_j^{(i)} \in \mathcal{C}^{(i)}$ to the participants $P_i \in \mathcal{P}$ ($i = 1, \ldots, n$)). Shares are sent to the participants via secure channels and are kept secret (only the dealer and the participant P_i know the share $c_j^{(i)}$).

2. *The Delegate (D) is trusted for the duration of a single transmission. Assume that there is a set $\gamma = \{P_{i_i}, \cdots P_{i_\ell}\}$ authorized of participants who want to authorize D to send a single source on their behalf. (D may choose any source state that it wishes for this single transmission.)*

   The delegate collects shares from the participants and uses the function

   $$f_D : \underbrace{\mathcal{C} \times \mathcal{C} \times \ldots \times \mathcal{C}}_{\ell} \rightarrow \mathcal{E}$$

   to compute the key $e \in \mathcal{E}$. This authorizes D to send any source state to the receiver by using the authentication matrix to select a codeword that is valid under the computed key e.

   For every encoding rule $e \in \mathcal{E}$ and for all authorized subsets of participants

   $$\gamma = \{P_{i_1}, \ldots, P_{i_\ell}\} \in \Gamma,$$

   the functions f_{DE} and f_D satisfies

   $$f_D(f_{DE}(e, P_{i_1}), \ldots, f_{DE}(e, P_{i_\ell})) = e.$$

   But if γ is unauthorized

   $$f_D(f_{DE}(e, P_{i_1}), \ldots, f_{DE}(e, P_{i_\ell})) \neq e$$

   with a high probability.

3. *Every ν_i-tuple belonging to $\prod_{P_j \in \Gamma_i} \mathcal{C}^{(j)}$, where Γ_i is a minimal authorized set of size ν_i, uniquely determines an $e \in \mathcal{E}$. Conversely, for every $e \in \mathcal{E}$ and every minimal authorized set Γ_i, the dealer DE can generate a unique ν_i-tuple of shares $c_{j_1}^{(i_1)} \cdots c_{j_{\nu_i}}^{(i_{\nu_i})}$ for participants in Γ_i such that $c_{j_\ell}^{(i_\ell)} = f_{DE}(e, P_\ell)$ and $f_D(c_{j_1}^{(i_1)} \cdots c_{j_{\nu_i}}^{(i_{\nu_i})}) = e$.*

Knowing the key, the receiver can check if the attached tag matches the tag in the authorization matrix.

It is clear that DD-systems can be seen as a family of secret sharing schemes that have in addition property (3). This property links a secret sharing scheme with an authentication matrix. Property (3) is essential and requires:

- that every ν_i-tuple uniquely defines an encoding rule;
- that every encoding rule specifies a unique ν_i-tuple.

If the first condition above is not satisfied then the delegate is unable to uniquely identify the encoding rule and the corresponding codeword. If the second condition is disregarded then some of ν_i-tuples are more probable then others. Giving an outsider a more efficient attack on the system. It is essential to keep the knowledge of any potential attackers as small as possible – this means that the probability distributions for the ν_i-tuples and encoding rules must be uniform.

DD systems in the minimal representation have the property that the dealer can create shares in two different ways. Firstly, DE follows the DD system definition, i.e. selects e and computes the shares. Secondly, the dealer chooses shares randomly with a uniform probability distribution from $\mathcal{C}^{(i)}$ for every participant P_i and later computes the secret encoding rule.

We note that in DD systems delegation is valid for only one transmission. This is similar to what happens in unconditional authentication systems in which protection is guaranteed for one transmission, or in secret sharing systems where the shares can only be used once. In Section 5 we extend this model to include multiple delegation.

Properties of DD systems for general access structures:

(P1) Let $\Gamma_v \subset \Gamma$ be an authorized set of participants and their (secret) shares. Then the participants are able to compute a unique encoding rule from the authentication matrix A.

(P2) Each participant P_i, $i = 1, \ldots, n$, knows a share $c_j^{(i)}$ from a collection $\mathcal{C}^{(i)}$ of all possible shares assigned to P_i. The collection $\mathcal{C}^{(i)} = \{c_1^{(i)}, \ldots, c_{\beta_i}^{(i)}\}$ is assumed to be public. This is a consequence of the assumption that the DD system is public except for the secret key and shares.

The following relates the numbers of shares in $\mathcal{C}^{(i)}$ to the number of encoding rules E.

Proposition 3.1 *If a DD system is defined for n participants P_i, $i = 1, \ldots, n$, whose corresponding sets of shares are $\mathcal{C}^{(i)} = \{c_1^{(i)}, \ldots, c_{\beta_i}^{(i)}\}$, then for every authorized set of participants $\Gamma_j \subseteq \Gamma$,*

$$E \leq \prod_{P_i \in \Gamma_j} |\mathcal{C}^{(i)}| = \prod_{P_i \in \Gamma_j} \beta_i.$$

The numbers $E, \beta_1, \ldots, \beta_n$ are the security parameters of a DD system. $1/E$ is the probability of guessing the secret encoding rule (the key) by an outsider. Note that this probability may be different for the holder of a secret share. Similarly, $1/\beta_i$ is the probability of guessing

the correct shares of P_i by an outsider. Again these probabilities can differ if the attacker is one of the participants or a collection of participants which does not belong to the access structure.

Consider the largest unauthorized collection of participants (not belonging to the access structure). Note that for this collection, the probability of guessing the secret is always equivalent to the probability of guessing a single share of one of the remaining participants.

4 Properties of DD-systems

To each $\mathcal{C}^{(i)}$ of size β_i we assign an index set I_i. We note that a collection of indices from I_i uniquely determines the corresponding collection of shares in $\mathcal{C}^{(i)}$.

We assume $I_i = \{1, 2, \cdots, \beta_i\}$ and $c_j^{(i)} \in \mathcal{C}^{(i)}$ is indexed by $j \in I_i$.

The design of a DD system is equivalent to solving an instance of the following general Indexing Problem. (In the rest of this paper we assume the minimal form for Γ.)

Indexing Problem in DD.

Given a collection of index sets I_E, $I_1, \ldots, I_n$ and an access structure Γ, how should indices from the sets $I_1, \ldots, I_n$ be chosen so as to satisfy the following conditions?

For each index $i \in I_E$ and every minimal authorized set Γ_j, there is a unique collection of indices in $\prod_{P_u \in \Gamma_j} I_u$ which selects that index, and conversely, for each ν_j-tuple of indices from $\prod_{P_u \in \Gamma_j} I_u$, where Γ_j is a minimal access set, there is a unique index in I_E.

The Indexing Problem (IP) can be characterized by the symbol

$$< \Gamma, n, \beta_1, \ldots \beta_n, E >$$

where Γ is the access structure, n is the number of participants, β_i is the number of elements in the set of indices assigned to P_i ($\beta_i = |\mathcal{C}^{(i)}|$) and E is the number of encoding rules.

For a general access structure there may be no solution, a unique solution or several solutions for the instance. If we assume that Γ is a (t, n)-access scheme, i.e. each set of participants of size t is authorized, the following is true.

Proposition 4.1 *Let Γ be a (t, n) access structure, then a solution to the Indexing Problem satisfies $\beta = \beta_1 = \cdots = \beta_n$.*

Proof We outline the proof. This is true because for an arbitrary set of indices, every t-tuple has to point to a unique index I_E. □

Threshold Indexing Problem in DD.

Given a collection of indices $\beta = \beta_1 = \cdots = \beta_n$, and a (t, n)-access structure how should the indices of $I_1, \ldots I_n$ be arranged so as to satisfy the following?

Every t-tuple from $I_{i_1} \times \cdots \times I_{i_t}$ uniquely determines an element of I_E, and every element of I_E together with a set $\{i_1, \cdots i_t\}$ uniquely determines a t-tuple from $I_{i_1} \times \cdots \times I_{i_t}$.

The Threshold Indexing Problem (TIP) can be specified by the symbol

$$< (t, n), \beta, E >$$

where (t, n) is the threshold access structure, β is the number of indices assigned to each participant and E is the number of encoding rules.

4.1 Threshold indexing problem

The indexing which solves an instance of the TIP Problem, can be characterized by the following corollary.

Corollary 4.1 *Assume that the dealer selects indices randomly and with uniform probability distribution, and there is a solution to the instance* $< (t,n), \beta, E) >$. *Then the probability of guessing the correct index by a participant (the key in the authentication matrix) is* $1/\beta^{t-1}$. *The probability of guessing the index of any other participant is* $1/\beta$. *The probability of guessing the authentication matrix index by an outsider is* $1/\beta^t$ *and the probability of guessing any participants index is* $1/\beta$.

The following examples illustrate a couple of solutions to the TIP.

Example

Consider the system with 3 participants (transmitters) A, B, and C whose sets of indices (shares) have two elements each. The access structure is $\Gamma = \{\{A,B\},\{A,C\},\{B,C\}\}$, a (2,3)-access scheme. This instance is denoted by $< (2,3),2 >$. The sets of indices (shares) are: $C^{(A)} = \{c_1^{(A)}, c_2^{(A)}\}$, $C^{(B)} = \{c_1^{(B)}, c_2^{(B)}\}$, $C^{(C)} = \{c_1^{(C)}, c_2^{(C)}\}$. Every pair of indices (shares) from two different sets should generate the unique index (encoding rule) in the matrix A. It is not hard to conclude that the shares have to be arranged as follows

$$\begin{aligned} e_1 &= (c_1^{(A)}, c_1^{(B)}, c_2^{(C)}) \\ e_2 &= (c_1^{(A)}, c_2^{(B)}, c_1^{(C)}) \\ e_3 &= (c_2^{(A)}, c_1^{(B)}, c_1^{(C)}) \\ e_4 &= (c_2^{(A)}, c_2^{(B)}, c_2^{(C)}). \end{aligned}$$

Note that each pair of participants can generate the index of the authentication matrix A (in this case the index is the scheme generated by two shares).

The next example is an instance of a $< (2,4),3 >$. In this case the access structure is equivalent to $\Gamma = \{\{A,B\},\{A,C\},\{A,D\},\{B,C\},\{B,D\},\{C,D\}\}$. One possible indexing of the authentication matrix A is given below:

	A	B	C	D
1	1	1	3	2
2	1	2	1	3
3	1	3	2	1
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	1	1
8	3	2	2	2
9	3	3	3	3

where the first column specifies all nine encoding rules, the integer i in the column x stands for $c_i^{(x)}$ ($i = 1,2,3$ and $x = A,B,C,D$). Any two shares uniquely determine the index (encoding rule) of the authentication matrix. (Note that $< (2,4),2 >$ has no solution.)

In general a solution to an indexing problem can be represented by an $E \times n$ matrix A, called the *indexing matrix*, where rows 1 to E correspond to the E elements of I_E and the n columns correspond to P_1 to P_n. The rows of A determine the assignment of indices to the participants.

Definition 4.1 *An orthogonal array $OA_\lambda(t,n,v)$ is an array with λv^t rows, each row of size k, from the elements of set X of v symbols, such that in any t columns of the array every possible v^t ordered sequence of elements of X occurs in exactly λ rows. Usually t is referred to as the strength of OA.*

The following proposition follows from definition of indexing problem.

Proposition 4.2 *In $a < (t,n), \beta, E >$ threshold indexing problem, the indexing array is an orthogonal array $OA(t,n,\beta)$.*

Corollary 4.2 *In $a < (t,n), \beta, E >$ threshold indexing problem $E = \beta^t$.*

4.2 Modified Threshold indexing problem

The indexing problem can be modified so as to improve our chances of realizing a given (t,n)-access structure. This course is motivated by the combinatorial constraint that $OD(t,n,v)$ do not exist for every (t,n)-access structure.

In applications we can relax the requirement of definition 4.1. The resulting combinatorial structure is called an *ordered design* $OD_\lambda(t,n,v)$. These designs are described by Teirlinck in [13]. In an ordered design $OD_\lambda(t,n,v)$, the requirement is that only the ordered t-tuples with no repeated entries appear in λ rows. We are interested in ordered designs where $\lambda = 1$.

We observe that the existence of an $OD(t,n,v)$ is equivalent to the solution of the modified threshold indexing problem (MTIP) for a (t,n)-access structure. The parameters of an $OD(t,n,v)$ relate to the MTIP as follows. The number of encoding rules E, is the number of rows of $OD(t,n,v)$, while (t,n) corresponds to a (t,n)-access structure and v is the number of tags assigned to each participant.

Corollary 4.3 *Assume that the dealer selects indices randomly and with uniform probability distribution, and an $OD(t,n,v)$ is used to realize a solution to the MTIP for an (t,n)-access structure. Then the probability of guessing the correct index by a participant (the key in the authentication matrix) is $1/n \cdots (n-t+1)$. The probability of guessing the index of any other participant is $1/n$. The probability of guessing the authentication matrix index by an outsider is $1/E$, where E is the number of rows of $OD(t,n,v)$, and the probability of guessing any participants index is $1/n$.*

Consider the following example of an $OD(2,3,3)$.

	A	B	C
1	0	1	2
2	1	2	0
3	2	0	1
4	0	2	1
5	2	1	0
6	1	0	2

This $OD(2,3,3)$ solves the MTIP for a $(2,3)$-access structure. For $(2,k)$-access structures we have the following.

Proposition 4.3 *Let C_p be a cyclic group of prime order p. For each element $0,1,\ldots p-1$ in C_p, let the rows of a $(p-1)\times p$ matrix A, be the cyclic permutations of the elements of C_p induced by $1,\ldots,p-1$ in the regular representation of C_p. Then A is an $OD(2,p,p)$.*

Proof The proof is given in the full version of the paper. □

$OD(2,n,n)$ also exist when $n = p^i$ is a prime power, but $OD(t,k,k)$ do not exist when when $t > 2$. We can also obtain an $OD(k-2,n,n)$ from the alternating groups A_n for $n \geq 3$. (A_n acts naturally on {1,...,n}.) This solves the MTIP for $(n-2,n)$-access structures, in this case the authentication matrix has $E = n!/2$ rows.

We remark that ordered designs $OD(2,p,p)$ arising from Proposition 4.3 solve the MTIP for $(2,n)$-access structures as follows. Given a $(2,n)$-access structure, let p be the smallest prime greater than n. Introduce 'dummy' participants: $P_{n+1}, P_{n+2}, \ldots, P_p$. By Proposition 4.3 we obtain a $OD(2,p,p)$, with participants $P_1, \ldots, P_p$. The solution to the MTIP for this $(2,n)$-access structure is obtained by restricting the $OD(2,p,p)$ to the columns indexed by $P_1, \ldots, P_n$.

5 General construction of DD systems using cumulative arrays

We can define a cumulative array for the access structure Γ in the same way as in the theory of secret sharing, (see [11]).

Definition 5.1 *A cumulative array CA_Γ for the access structure Γ is a pair: the collection of sets of indices $I = \{I_1, \ldots, I_m\}$ and the index function*

$$\mathcal{I} : \mathcal{P} \to 2^I$$

which assigns a subset of indices to each participant in such a way that any authorized subset $\mathcal{A} \subset \Gamma$ of participants knows the whole collection of indices I and any unauthorized subset of participants does not.

For any access structure Γ, it is possible to create an infinite sequence of cumulative arrays. However there is always at least one which involves the smallest number of indices. This cumulative array is called *minimal* and it determines the parameter m.

Algorithm for the dealer.

- Given the access structure Γ, the dealer computes the minimal cumulative array AC_Γ with indices $I = \{I_1, \ldots, I_m\}$. Assuming that the index set is $I_i = \{x^{(i)_1}, \ldots, x^{(i)}_{j_i}\}$ for $i = 1, \ldots, m$. The array AC_Γ is public.

- The dealer selects at random sequence of indices

  $$x^{(1)}_{k_1} \in I_1, x^{(2)}_{k_2} \in I_2, \ldots, x^{(m)}_{k_m} \in I_m,$$

  creates the encoding rule

  $$e = x^{(1)}_{k_1} \parallel x^{(2)}_{k_2} \parallel \ldots \parallel x^{(m)}_{k_m}$$

  and sends it to the receiver via secure channel.

- Next the dealer composes shares for participants

  $$c_i = x^{(i_1)}_{k_{i_1}} \parallel x^{(i_2)}_{k_{i_2}} \parallel \ldots \parallel x^{(i_r)}_{k_{i_r}}$$

  where $x^{(i_\ell)}_{k_{i_\ell}} \in I_\ell$ and $I_\ell \subset \{I_{i_1}, \ldots, I_{i_r}\} = \mathcal{I}(P_i)$. The shares c_i $(i = 1, \ldots, n)$ are communicated to corresponding participants via secure channels.

Algorithm for the delegate.

- The delegate collects shares from participants. If the collaborating subset belongs to the access structure, the delegate is able to retrieve the encoding rule e from the shares. On successful computation of the encoding rule, a single codeword is constructed and sent to the receiver via insecure channel.

Proposition 5.1 *A minimal cumulative array $\mathcal{AC}_\Gamma$ for the access structure Γ on n participants generates a solution for an instance of the Indexing Problem*

$$< \Gamma, n, \beta_1, \ldots, \beta_n >$$

where β_i gives the number of shares assigned to the participant P_i and

$$\beta_i = \prod_{I_j \in \mathcal{I}(P_i)} |I_j|.$$

Cumulative arrays can also be used for threshold indexing. In the previous section we discussed this and assumed that that the sets of indices used by participants do not overlap. Here the cumulative array is used to determine the overlap of indices. The probabilities of guessing correct index (encoding rule) by participants are now determined by the access structure.

Example 1.

If $\Gamma = \{\{A, B\}, \{B, C\}, \{C, D\}\}$ then its minimal cumulative array $\mathcal{AC}_\Gamma$ is:

	I_1	I_2	I_3
A	1	0	0
B	0	1	1
C	1	1	0
D	0	0	1

.

This means that A is assigned $\{I_1\}$ (A's shares belong to I_1), B the set $\{I_2, I_3\}$ (B's shares are from $I_2 \times I_3$), C the set $\{I_1, I_2\}$ (C's shares are from $I_1 \times I_2$), and D the set $\{I_3\}$. Note that every $\mathcal{A} \subset \Gamma$ can retrieve the index $e \in I_1 \times I_2 \times I_3$) and any unauthorized subset cannot.

Example 2.

Assume that $\Gamma = \{\{A, B\}, \{A, C\}, \{B, C\}\}$. The corresponding cumulative array is:

	I_1	I_2	I_3
A	1	1	0
B	1	0	1
C	0	1	1

.

Participants are assigned the following sets of shares:

A	B	C
$I_1 \times I_2$	$I_1 \times I_3$	$I_2 \times I_3$
00	00	00
$\rightarrow$ 01	$\rightarrow$ 01	01
10	10	10
11	11	$\rightarrow$ 11

.

Current values of the shares are indicated by ($\rightarrow$). The authentication matrix is indexed by $e = (e_1, e_2, e_3)$ where $e_i \in I_i = \{0, 1\}$ for $i = 1, 2, 3$. It has 8 rows. The current index (encoding rule) in it is $e = (011)$. An outsider can guess the share for P_i ($i = 1, 2, 3$) with probability 1/4 and the index (encoding rule) with the probability 1/8. However any participant can guess the other participants indices with probability 1/2.

In general, the DD systems can be divided into two classes:

1. Ideal DD systems. The underlying secret sharing scheme is ideal for every encoding rule.

2. Non-ideal DD systems. This class includes all systems which are not ideal.

Ideal DD systems provide the best security in terms of the probability of guessing the secret encoding rule (key), or guessing the share of each participant P_i which is $1/E$. This also means that the lengths of all shares and the secret encoding rule are the same.

6 Multiple Delegation

The delegation systems discussed in previous sections allow the delegate to construct a single authentic message. Suppose now we require that delegation remains valid for m transmissions such that the security of the system be retained. One way of achieving this goal is to use an authentication matrix that provides high protection against spoofing of all orders $i, for\ 0 \leq i \leq m$, and allow the delegate and the receiver to use the same key for m transmissions. With this approach the protective properties of the authentication matrix play a crucial role, and for large m the construction of such matrices is a very difficult task. For example, if we insist on having perfect protection in all transmissions we would require an authentication matrix that provides perfect protection for spoofing of all orders i, $0 \leq i \leq m$. This is equivalent to construction of an orthogonal array of strength $m+1$. An alternative is to use an authentication matrix that only provides protection against impersonation and change the key after each transmission. With this approach the cost of key exchange becomes prohibitive for large m

In this section we introduce ideal families of threshold schemes. These can be used for secure multiple delegation systems while reducing the cost of key exchange compared to the m-fold requirement of a standard DD-system.

6.1 Ideal DD systems

In an ideal DD system the underlying secret sharing scheme is ideal. A secret sharing scheme is *ideal* if it is *perfect* and the numbers of elements in the share sets $|C^{(i)}| = |\mathcal{E}|$ for all $i = 1, \ldots, n$. In a perfect secret sharing scheme the knowledge of the shares of an unauthorized set of participants does not decrease the entropy of the secret key (see [7], [12]).

It is known [12], that some access structures cannot be realized as ideal secret sharing schemes. To simplify our discussion, we will assume that the access structure Γ we are considering is a (t, n)-access structure. These can always be implemented as an ideal sharing scheme.

Definition 6.1 *A family of threshold systems (TS family) is defined by an* $(m \times n)$ *matrix of shares* $[c_{i,j}]$ *such that:*

1. *Any row* $(c_{i,1}, c_{i,2}, \ldots, c_{i,n})$ *represents an instance of threshold scheme* $TS_{r_i} =< t_i, n >$ *where* $i = 1, \ldots, m$.

2. *Any column* $(c_{1,j}, c_{2,j}, \ldots, c_{m,j})$ *represents an instance of* $TS_{q_j} = < t_j, m >$ *where* $j = 1, \ldots, n$.

A TS family whose all row and column threshold schemes are ideal, is called *ideal threshold scheme family* or a *ITS family.*

6.2 Properties of ITS family

Note that an ideal (t, n)-threshold scheme has the property that any subset of t shares is a collection of independent random variables.

Lemma 6.1 *(Karnin, Greene, Hellman [9]) Given an ideal threshold scheme $< t, n >$ with n shares $C_n = (c_1, \ldots, c_n)$ (where $c_i \in \mathcal{C}$ for $i = 1, \ldots, n$), then knowledge of any collection of $(t-1)$ shares $C_{t-1} = (c_1, \ldots, c_{t-1})$ and the secret $e \in \mathcal{E}$ uniquely identifies the remaining shares.*

Proof Without loss of generality, we assume that $C_{t-1} = (c_1, \ldots, c_{t-1})$ and $C_t = (c_1, \ldots, c_t)$. As we are dealing with ideal threshold scheme, we have

$$\begin{aligned} H(\mathcal{E} \mid C_{t-1}) &= H(\mathcal{E}); \\ H(\mathcal{E} \mid C_t) &= 0; \\ H(\mathcal{E}) &= H(\mathcal{C}). \end{aligned}$$

Any collection of t or less shares can be considered as a collection of independent and uniform random variables so

$$\begin{aligned} H(C_t) &= t\, H(\mathcal{C}); \\ H(C_{t-1}) &= (t-1) H(\mathcal{C}), \end{aligned}$$

where $\mathcal{C}$ is the random variable used repeatedly to generate the shares. It is obvious that

$$\begin{aligned} H(c_t \mid C_{t-1}, \mathcal{E}) &= H(C_t, \mathcal{E}) - H(C_{t-1}, \mathcal{E}) \\ &= H(C_t) - H(\mathcal{E}) - H(C_{t-1}) = 0. \end{aligned}$$

□

The following are corollaries of Lemma 6.1.

1. The knowledge of t shares is suffices not only to determine the secret but the rest of the shares as well.
2. The secret $e \in \mathcal{E}$ can be seen as one of the shares, and any collection of $(t-1)$ shares together with the secret determines the rest of the shares.
3. The ideal threshold scheme is uniquely defined by any subset of t shares (or equivalently $t-1$ shares plus the secret).

Two ideal threshold schemes $TS_1 = (t_1, n_1)$, $TS_2 = (t_2, n_2)$ are independent if any subset C_1 of shares of TS_1 is statistically independent from any subset C_2 of shares of TS_2. This condition can stated in terms of entropy as

$$\forall_{C_1 \subset \mathcal{C}^{n_1}} \forall_{C_2 \subset \mathcal{C}^{n_2}} H(C_1 \mid C_2) = H(C_1) \text{ and } H(C_2 \mid C_1) = H(C_2).$$

For any row (or column) of an ITS family, there is a sequence of t_{r_i} (t_{q_i} for columns) of independent random variables, where t_{r_i} and t_{q_i} are the threshold values of the row and column threshold schemes, respectively.

Any threshold scheme has a secret encoding rule associated with it. In an ITS family, we can associate a row of secrets $e_r = (e_{q_1}, \ldots, e_{q_n})$ for the corresponding column threshold schemes. We can also define a column of secrets $e_c = (e_{r_1}, \ldots, e_{r_m})$ for the corresponding row threshold schemes. It is easy to prove that both the row secret e_r and the column secret e_c are threshold schemes.

Definition 6.2 *The kernel of ITS family (ITSF) is a maximal subset of independent shares.*

Theorem 6.1 *Let $ITSF_{m\times n}$ be a family of threshold schemes defined by matrix $m \times n$ such that any subset of u row threshold schemes create a collection of independent TS and any subset of v column TS is a collection of independent TS. Then the kernel of the $ITSF$ is a sub-matrix $[s_{i,j}]$ $(i = 1, \ldots, u,\ j = 1, \ldots, v)$. The threshold values for all row TS is v and for all column TS is u.*

Proof As any subset of u row TS is a collection of u independent TS, any column of shares $(c_{1,i}, c_{2,i}, \ldots, c_{u,i})$ constitutes a collection of independent shares. If this collection is maximal (i.e. any set of $u + 1$ shares is dependent), the rest of the column shares $(c_{u+1,i}, \ldots, c_{m,i})$ are dependent on the first u shares. Thus by Lemma 6.1, any column TS has its threshold value equal to u. The same applies to row TS whose threshold values are the same and equal to v. Any v shares generate each row TS so $ker_r = [c_{i,j}]$, where $i = 1, \ldots, m$ and $j = 1, \ldots, v$, are all shares which are 'row' independent. Any u shares generate each column TS so $ker_c = [c_{i,j}]$, where $i = 1, \ldots, u$ and $j = 1, \ldots, n$, are all shares which are 'column' independent. Therefore, the kernel is $ker = ker_r \cap ker_c$, which is a $u \times v$ submatrix of shares. □

¿From Lemma 6.1 and Theorem 6.1, we can draw the following conclusions:

1. For any $ITSF_{m\times n}$, there is a column of secrets $e_c = (e_{r_1}, \ldots, e_{r_m})$ and a row of secrets $e_r = (e_{c_1}, \ldots, e_{c_m})$.
2. Any subset of u secrets of e_c is enough to recreate the whole sequence of e_c.
3. Any subset of v secrets of e_r is enough to recreate the whole sequence of e_r.

The relation between the column secret e_c and the row secret e_r is described by the following theorem.

Theorem 6.2 *Let $ITSF_{m\times n}$ be a family of ideal TS with the kernel $[c_{i,j}]$ $(i = 1, \ldots, u,\ j = 1, \ldots, v)$. Then*

$$| e_c \cap e_r | = 1$$

where $e_c \in \mathcal{E}_c$, $e_r \in \mathcal{E}_r$, and

$$\begin{aligned} H(\mathcal{E}_c \mid \mathcal{E}_r) &= (u-1)\, H(\mathcal{E}); \\ H(\mathcal{E}_r \mid \mathcal{E}_c) &= (v-1)\, H(\mathcal{E}). \end{aligned}$$

Proof Note that e_c and e_r can be considered as a column row respectively, of the ITSF share matrix $[c_{i,j}]$ $(i = 1, \ldots, m,\ j = 1, \ldots, n)$. Obviously any column and any row intersect and the intersection contains a single share. As any column in the matrix consists of u independent shares (i.e. $H(\mathcal{E}_c) = u\, H(\mathcal{E})$), the conditional entropy is $H(\mathcal{E}_c \mid \mathcal{E}_r) = (u-1)\, H(\mathcal{E})$. □

6.3 Multiple delegations based on ITSF

An ITS family provides a convenient approach to the design of ideal DD systems. The main advantage of an ITS family is that the dealer can control the internal structure of such a scheme and knows interrelations among the components (row and column threshold schemes). Also in the initialization stage, the dealer can distribute shares in one single transmission.

Assume that we have an ITS family defined by $[c_{i,j}]_{m\times n}$. Then instead of a single share, the dealer in our ideal DD system can assign a column of the matrix $[c_{i,j}]$ (a j-th column threshold scheme) to each of the n participants. The rows of the ITSF specify threshold schemes for subsequent m instances of time. Such ideal DD systems are called *multiple* DD systems or MDD systems.

Let us now consider a MDD system based on ITS family whose kernel is a $u \times v$ rectangle. For such system the following properties are easy to prove.

1. All shares are represented by column threshold schemes of the same threshold value u. So every participant can reveal up to $u-1$ single points from their shares without compromising their other shares.

2. Threshold schemes for all time instances have the same threshold value v.

3. The knowledge of u secret encoding rules (for u rows of ITSF) determines the rest of the secrets.

4. Delegate can be asked to perform multiple authorization up to $u-1$ for a single participant. If the collaborating groups of participants do not overlap or the overlap is small, ITSF can be used up to m times.

Suppose we have a MDD system based on an ITSF whose kernel is a triangle: $ker = [c_{i,j}]$; $i,j = 1,\ldots,u$ and $i+j \leq u+1$. We can create rows from the kernel. The first row is a $TS(u,n)$, the second corresponds to $TS(u-1,n)$, etc.; the u-th row is a $TS(1,n)$. The first column of the ITSF matrix – the the first participant's share – is obviously $TS(u,m)$. The remaining columns also correspond to $TS(u,m)$. To clarify this, note that first u rows are independent TS. So any column contains u independent random variables, thus any column has the same threshold value u. By the same token, the rest of the rows are $TS(u,n)$. Hence, dynamic secret sharing schemes with triangle kernels have the following properties:

1. Threshold values can be reduced (increased) by one between subsequent time instances.

2. Threshold values stay the same for all time instances (all rows) if u columns are generated first. But participants are given TS (shares) of different threshold values (different degree of vulnerability on disclosure of single points of their shares).

3. Threshold values can vary for each time instance (row) TS and for each participant share (column) and they take on values from the interval $(u-j+1,u)$ where j specifies the row or the column of the ITSF matrix and $j = 1,\ldots,u$. The variations relate to the algorithm which is used to generate the $[c_{i,j}]$ matrix of ITSF having the kernel (the order of creating rows and columns from the kernel).

4. As the delegate should be unable to compute shares for subsequent time instances from the previous authorizations, it is necessary to distribute the columns (shares) with high threshold values to participants who are liable to use the system more frequently. While participants who will use the system infrequently, should be assigned columns (shares) with lower thresholds.

MDD systems provide the best possible protection against an outsider and a participant who would like to cheat. For both attackers, the probability of guessing the secret (encoding rule), or a share of the other participants is $1/E$.

7 Conclusions

Distributed delegation systems allow a group of participants to transfer the authority of computing a tag to a trusted delegate. The delegate always succeeds in computing the correct tag if the group of transmitters belongs to the access structure. The computed tag is attached to the message and forwarded to the receiver. Knowledge of the secret key, enables the receiver to check the validity of the message and the tag.

References

[1] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. Technical report, DEC Research Report, No 70, Feb 28, 1991.

[2] M.N. Wegman, J.L. Carter. New hash functions and their use in authentication and set equality. J. Computer and System Sciences, Vol. 22, 1981, pp 265-279.

[3] D. Chaum and E. van Heyst. Group signature. In *Advances in Cryptology - EUROCRYPT'91, D.W. Davies (ED), Lecture Notes in Computer Science, Vol.547*, pages 257–265. Springer Verlag, 1991.

[4] R.A. Croft and S.P. Harris. Public-key cryptography and re-usable shared secrets. In *Cryptography and Coding, H. Beker and F. Piper (Eds)*, pages 189–201. Clarendon Press, 1989.

[5] Y. Desmedt. Society and group oriented cryptography: a new concept. In *Advances in Cryptology, Proceedings of CRYPTO'87, Ed C. Pomerance, Lecture Notes in Computer Science, Vol.293*, pages 120–127. Springer-Verlag, 1988.

[6] Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In *Avances in Cryptology - CRYPTO'91, Lecture Notes in Computer Science, Vol. 718, J.Feigenbaum (Ed.)*, pages 457–569. Springer-Verlag, 1992.

[7] G.J. Simmons (Ed). *Contemporary Cryptology - The Science of Information Integrity.* IEEE Press, New York, 1992.

[8] T. Hwang. Cryptosystem for group oriented cryptography. In *Advances in Cryptology - EUROCRYPT'90, I. Damgard (ED), Lecture Notes in Computer Science, Vol.473*, pages 353–360. Springer Verlag, May 1990.

[9] E.D. Karnin, J.W. Greene, and M.E. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, IT-29:35–41, 1983.

[10] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[11] G.J. Simmons, W. Jackson, and K. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1:71–88, 1991.

[12] D.R. Stinson. An explication of secret sharing schemes. *Designs, Codes and Cryptography*, 2:357–390, 1992.

[13] L. Teirlinck. Large sets of disjoint designs and related structures. In *Contemporary Design Theory, A Collection of Surveys, J.H Dinitz and D.R. Stinson (Eds.)*, pages 561–592. John Wiley and Sons, Inc. New York, 1992.