General Short Computational Secret Sharing Schemes

Philippe Béguin[1] and Antonella Cresti[2,*]

[1] Laboratoire d'Informatique [**]
Ecole Normale Supérieure, 75230 Paris Cédex 05, France
e-mail: beguin@truffe.ens.fr

[2] Dipartimento di Scienze dell'Informazione
Università di Roma "La Sapienza", 00198 Roma, Italy
e-mail: antone@dsi.uniroma1.it

Abstract. A secret sharing scheme permits a secret to be shared among participants in such a way that only qualified subsets of participants can recover the secret. If any non qualified subset has absolutely no information about the secret, then the scheme is called perfect. Unfortunately, in this case the size of the shares cannot be less than the size of the secret. Krawczyk [9] showed how to improve this bound in the case of computational threshold schemes by using Rabin's information dispersal algorithms [14], [15].

We show how to extend the information dispersal algorithm for general access structure (we call access structure, the set of all qualified subsets). We give bounds on the amount of information each participant must have. Then we apply this to construct computational schemes for general access structures. The size of shares each participant must have in our schemes is nearly minimal: it is equal to the minimal bound plus a piece of information whose length does not depend on the secret size but just on the security parameter.

1 Introduction

Secret sharing is an important tool in security and cryptography. An important issue in secret sharing theory is the size of the share distributed, since the security of a system degrades as the amount of information that must be kept secret increases. A very strong requirement is that all qualified subsets of participants can reconstruct the secret but all other subsets obtain no information (in an information-theoretic sense) about the secret. These schemes are called *perfect* secret sharing schemes. Unfortunately, in this case the size of the shares cannot

[*] Part of this work was done while the author was visiting the Laboratoire d'Informatique of the Ecole Normale Supérieure, France.

[**] Supported by the Centre National de la Recherche Scientifique URA 1327.

L.C. Guillou and J.-J. Quisquater (Eds.): Advances in Cryptology - EUROCRYPT '95, LNCS 921, pp. 194-208, 1995.

be less than the size of the secret. However, the proof of this lower bound uses the notion of information-theoretic secrecy.

A natural question is whether one can do better for secret sharing if the notion of secrecy is computational, namely, against resource bounded adversaries: i.e. any qualified subset can reconstruct the secret but any other subset obtains no computational information about the secret. These schemes are called *computational* secret sharing schemes.

Krawczyk [9] proposed a computational m-threshold scheme, where m shares recover the secret but $m-1$ shares give no (computational) information on the secret, in which shares corresponding to a secret uniformly chosen in a set S are of size $\log |S|/m$ (where $|S|$ denotes the cardinality of the set S) plus a short piece of information whose length does not depend on the secret size but just on the security parameter. In our paper, $|X|$ denotes the cardinality of the set X, whereas in Krawczyk's one, $|x|$ denotes the size of x for $x \in X$.

The scheme of Krawczyk is very simple and combines in a natural way traditional (perfect) secret sharing schemes, encryption, and known information dispersal algorithms. It is provable secure given a secure (private key) encryption function.

A natural and open question is whether the space efficiency can be carried over more general access structures than just threshold schemes: one of the problems was to find an information dispersal algorithm for general access structures.

In this paper, we define information dispersal algorithms for general access structures; we show bounds on the size of pieces each participant must have and we give practical constructions of algorithms that reach these bounds. Then we apply these results to computational secret sharing schemes. We show how to realize computational secret sharing schemes for general access structures that are nearly optimal: the size of each share is equal to the minimal theoretical bound plus a piece of information whose length does not depend on the secret size but just on the security parameter.

2 Perfect Secret Sharing (PSS)

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be the set of participants. Denote by $\mathcal{A} \subseteq 2^{\mathcal{P}}$ the family of subsets of participants which we desire to be able to recover the file; $\mathcal{A}$ is called the *access structure*. It is reasonable to require that $\mathcal{A}$ be *monotone*, that is if $A \in \mathcal{A}$ and $A \subseteq A' \subseteq \mathcal{P}$, then $A' \in \mathcal{A}$.

If $\mathcal{A}$ is an access structure on $\mathcal{P}$, then $B \in \mathcal{A}$ is a *minimal* authorized subset if $A \notin \mathcal{A}$ whenever $A \subset B$. The set of minimal authorized subsets of $\mathcal{A}$ is denoted $\mathcal{A}^0$ and is called the *basis* of $\mathcal{A}$. $\mathcal{A}$ is uniquely determined as a function of $\mathcal{A}^0$, as we have $\mathcal{A} = \{B \subseteq \mathcal{P} : \exists A \subseteq B,\ A \in \mathcal{A}^0\}$. We say that $\mathcal{A}$ is the *closure* of $\mathcal{A}^0$ and write $\mathcal{A} = cl(\mathcal{A}^0)$.

Given an access structure $\mathcal{A}$, on a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of participants, let be K the space of secrets, and let $\{p_K(k)\}_{k \in K}$ be a probability distribution on K. Let a secret sharing scheme for secrets in K be fixed. For any participant $P_i \in \mathcal{P}$, let us denote by V_i the set of all possible shares given to participant P_i.

Given a set of participants $A = \{P_{i_1}, \ldots, P_{i_r}\} \subseteq \mathcal{P}$, where $i_1 < i_2 < \ldots < i_r$, denote by V_A the set $V_{i_1} \times \cdots \times V_{i_r}$. A secret sharing scheme for secrets in K and a probability distribution $\{p_K(k)\}_{k \in K}$ naturally induce a probability distribution on V_A, for any $A \subseteq \mathcal{P}$. Denote such probability distribution by $\{p_{V_A}(a)\}_{a \in V_A}$. Finally, denote by $H(K)$ the entropy of $\{p_K(k)\}_{k \in K}$ and by $H(V_A)$ the entropy of $\{p_{V_A}(a)\}_{a \in V_A}$, for any $A \in 2^{\mathcal{P}}$.

Following the information-theoretical approach of [6] and [4] we have the following definition.

Definition 1. Let $\mathcal{A}$ be an access structure on a set $\mathcal{P}$ of participants. We say that a secret sharing scheme for secrets in K is *perfect* for the access structure $\mathcal{A}$ on $\mathcal{P}$, if the following two properties hold:

1. *Any qualified subset can reconstruct the secret:* Formally, for all $A \in \mathcal{A}$, it holds $H(K|V_A) = 0$.
2. *Any non-qualified subset has absolutely no information on the secret:* Formally, for all $A \notin \mathcal{A}$, it holds $H(K|V_A) = H(K)$.

3 Information Dispersal Algorithms (IDA)

We analyze the problem of distributing pieces of a file f among a set of users in such a way that some predefined subsets of users can, pooling together their pieces, reconstruct the entire file f. An information dispersal algorithm differs from a secret sharing scheme as there are no restriction whatsoever about the sets which are not in $\mathcal{A}$. Rabin ([14],[15]) first considered the problem and introduced the *Information Dispersal Algorithms*. His schemes are intended for the distribution of a piece of information among n active processors, in such a way that the recovery of the information is possible in presence of m active processors, where m and n are parameters satisfing $1 \leq m \leq n$. The basic idea of his algorithms is to add to the information some amount of redundancy and then to partition it into n fragments, each transmitted to one of the parties. Reconstruction of f is possible out of m fragments. Information dispersal algorithms have several applications to secure and reliable storage of information in computer networks. Moreover they can be applied to fault-tolerant transmission of information and to communication between processors in parallel computers.

Subsequently, Naor and Roth [12], using integer linear programming techniques, proposed an information dispersal algorithm over arbitrary graphs. In their model, an arbitrary file f is distributed among the nodes of the graph in such a way that each node of the graph, by accessing the memory of its own and of its adjacent nodes, can reconstruct the contents of f. Their scheme can be applied to store files in distributed networks.

In this paper, we define information dispersal algorithms in a similar way than secret sharing schemes, using an information–theoretical approach. Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be the set of participants; we denote by $\mathcal{A}$ the access structure that is the subsets of participants which we desire to be able to recover the file: $\mathcal{A}$

is monotone. We define in a similar way respect to secret sharing schemes the *minimal authorized* subsets, the *basis* and the *closure* of $\mathcal{A}$.

If F is the set of files, $\{p_F(f)\}_{f \in F}$ a probability distribution on F, and an information dispersal algorithm for files in F is fixed, we define as $V_i, V_A, H(K)$ respectively $G_i, G_A, H(F)$.

Definition 2. Let $\mathcal{A}$ be an access structure on a set $\mathcal{P}$ of participants. We say that an algorithm Σ to distribute a file in F according with the probability distribution $\{p_F(f)\}_{f \in F}$ is an Information Dispersal Algorithm (IDA) if any qualified subset can reconstruct the file. Formally, for all $A \in \mathcal{A}$, it holds

$$H(F|G_A) = 0.$$

The following lemma holds.

Lemma 3. Let $\mathcal{A}$ be an access structure on a set $\mathcal{P}$ of participants. Any information dispersal algorithm for $\mathcal{A}$, for any $A \in \mathcal{A}$, must give to at least a participant $P_j \in A$ a fragment from a domain G_j such that $H(G_j) \geq H(F)/|A|$.

Proof. Let $A \in \mathcal{A}$. Consider the conditional mutual information $I(G_A; F)$. It can be written either as $H(G_A) - H(G_A|F)$ or as $H(F) - H(F|G_A)$. Hence, from (4) of Appendix A and from 1. of definition 2 we have

$$H(G_A) = H(F) + H(G_A|F) \geq H(F). \tag{1}$$

From (5) and (8) of appendix A it follows that $\sum_{P_i \in A} H(G_i) \geq H(G_A)$. So, from (1) one gets that there exists a participant P_j in A such that $H(G_j) \geq H(F)/|A|$. □

3.1 A Simple Information Dispersal Algorithm

We outline the following simple information dispersal algorithm $\Sigma(m, b)$, where b is the number of file fragments and m is the minimum number of fragments required to reconstruct the file. It is a simple version of Rabin's information dispersal algorithm. The algorithm is based on Reed Solomon erasure codes. The information $f \in F$ to be shared is first partitioned into m equal parts where each part is viewed as an element over a finite field (e.g. $GF(q)$, for a large enough q). These m elements are then viewed as coefficients of a polynomial of degree $m-1$, and the b fragments for distribution are obtained by evaluating this polynomial in b different points (we need $q \geq b$). Clearly the whole information can be reconstructed (by interpolation) from any m fragments.

Assuming the uniform probability distribution over the set of files F, we have $H(F) = \log|F| = m \log q$. Moreover, for all participant $P_i \in \mathcal{P}$ it holds $H(G_i) \leq \log|G_i| = \log q = \log|F|/m$.
Observe that we have the requirement $q \geq b$. This implies $\log|F| = m \log q \geq m \log b$. So when the parameters m and b are big the algorithm works for large files.

3.2 The Size of Pieces

The efficiency of any information dispersal algorithm, is computed regarding the size of pieces given to each participant. So, even in the case of general access structures, we are interested to minimize the size of fragments distributed to participants.

Information Rate

If we are interested in limiting the maximum size of fragments for each participant (i.e., the maximum quantity of information that must be given to any participant), then a worst-case measure of the maximum of $H(G_i)$ over all $P_i \in \mathcal{P}$ naturally arises. Analogously to definition of *information rate* for secret sharing schemes presented in [2], we give the following definition.

Definition 4. We define the *information rate* of an information dispersal algorithm Σ for the access structure $\mathcal{A}$, when the probability distribution on the set of files F is Π_F, as

$$\varrho(\mathcal{A}, \Pi_F, \Sigma) = \frac{H(F)}{\max\{H(G_i)\ :\ 1 \le i \le n\}}.$$

The following theorem holds.

Theorem 5. Let $\mathcal{A}$ be an access structure on a set $\mathcal{P}$ of participants. The information rate of any information dispersal algorithm Σ for $\mathcal{A}$ satisfies

$$\varrho(\mathcal{A}, \Pi_F, \Sigma) \le \varrho_{\max},$$

where $\varrho_{\max} = \min\{|A| : A \in \mathcal{A}^0\}$.

Proof. Let $A \in \mathcal{A}^0$ such that $|A| = \varrho_{\max}$. From Lemma 3, any information dispersal algorithm Σ for F must give to at least a participant $P_j \in A$ a fragment such that $H(G_j) \ge H(F)/\varrho_{\max}$. So, for any information dispersal algorithm Σ, $\max\{H(G_i)\ :\ 1 \le i \le n\} \ge H(G_j) \ge H(F)/\varrho_{\max}$. Hence,

$$\varrho(\mathcal{A}, \Pi_F, \Sigma) = \frac{H(F)}{\max\{H(G_i)\ :\ 1 \le i \le n\}} \le \varrho_{\max}.$$

□

Average Information Rate

In many cases it is preferable to limit the sum of the size of fragments given to all participants. In such a cases the arithmetic mean of the size of fragments for each participant is a more appropriate measure.

Definition 6. We define the *average information rate* of an information dispersal algorithm Σ for an access structure $\mathcal{A}$ when the probability distribution on the set of files F is Π_F, as

$$\tilde{\varrho}(\mathcal{A}, \Pi_F, \Sigma) = \frac{nH(F)}{\sum_{i=1}^{n} H(G_i)}.$$

Consider the following linear programming problem LP1.

$$\boxed{\begin{array}{lll} \text{Minimize} & M = \sum_{i=1}^{n} \alpha_i & \text{subject to:} \\ & \alpha_i \geq 0, & 1 \leq i \leq n \\ & \sum_{P_i \in A} \alpha_i \geq 1, & \forall A \in \mathcal{A}^0 \end{array}}$$

Let $M_{\min}^{(1)}$ the solution of the linear programming problem LP1. The following theorem holds.

Theorem 7. Let $\mathcal{A}$ be an access structure on a set $\mathcal{P}$ of participants. The average information rate of any information dispersal algorithm Σ for $\mathcal{A}$ satisfies

$$\tilde{\varrho}(\mathcal{A}, \Pi_F, \Sigma) \leq \frac{n}{M_{\min}^{(1)}}.$$

Proof. Let $A \in \mathcal{A}^0$. From Lemma 3, it follows that $\sum_{P_i \in A} H(G_i) \geq H(G_A) \geq H(F)$. Let $x_i = H(G_i)/H(F)$, for $1 \leq i \leq n$. We have $x_i \geq 0$, and $\sum_{P_i \in A} x_i \geq 1$, $\forall A \in \mathcal{A}^0$. Then $\sum_{i=1}^{n} x_i \geq M_{\min}^{(1)}$. Hence,

$$\tilde{\varrho}(\mathcal{A}, \Pi_F, \Sigma) = \frac{n}{\sum_{i=1}^{n} x_i} \leq \frac{n}{M_{\min}^{(1)}}.$$

□

3.3 General Information Dispersal Algorithms

The schemes in this section are obtained supposing the uniform probability distribution on F: hence $H(F) = \log|F|$. Let $\mathcal{A}$ be an access structure on a set $\mathcal{P}$ of participants. For all $P_i \in \mathcal{P}$, let p_i, q_i be some positive integers such that

$$\sum_{P_i \in A} \frac{p_i}{q_i} \geq 1 \qquad \forall A \in \mathcal{A}^0. \tag{2}$$

Let m be the least common multiple of $q_1, \ldots, q_n$, let $x_i = p_i/q_i$, for $i = 1, \ldots, n$, and let $b = \sum_{i=1}^{n} m \cdot \frac{p_i}{q_i}$. We assume the information dispersal algorithm $\Sigma(m, b)$ described in section 3.1 which works for parameters b (number of file fragments) and m (number of required fragments to reconstruct the file). We explain later how to choose the values x_i, for $i = 1, \ldots, n$, in order to optimize the information rate or the average information rate of the scheme.

Distribution Scheme

- Using $\Sigma(m, b)$ partition the file $f \in F$ into b fragments, $f_1, \ldots, f_b$.
- Assign to each participant P_i, $m \cdot x_i$ distinct fragments $f_1^{(i)}, \ldots, f_{m \cdot x_i}^{(i)}$ (this is always possible since $\sum_{i=1}^{n} m \cdot x_i$ is equal to b, the number of available fragments).

The fragment of each participant P_i consists on $g_i = (f_1^{(i)}, \ldots, f_{m \cdot x_i}^{(i)})$, for $i = 1, \ldots, n$.

Reconstruction Scheme

- Each set of participants A in the access structure collect their fragments.
- Using $\Sigma(m, b)$ reconstruct f out of the collected values.

Proposition 8. The above scheme constitutes an information dispersal algorithm for the access structure $\mathcal{A}$.

Proof. For all $A \in \mathcal{A}^0$, from condition (2) there holds $\sum_{P_i \in A} m \cdot x_i \geq m$. From this fact, and from the properties of the algorithm $\Sigma(m, b)$ derives the feasibility for a set of participants A to reconstruct the file f out of the fragments. □

3.4 How to Optimize the Information Rate

We now show how to choose the values x_i, for $1 \leq i \leq n$ in order to maximize the information rate. We propose two techniques. Both of them are optimal as they reach the lower bound proved in section 3.2. Moreover, the algorithm obtained applying the second technique gives to participants fragments no longer than necessary.

First Technique

We propose the following simple method to choose the values $x_i = p_i/q_i$, for all $i = 1, \ldots, n$. For all participants $P_i \in \mathcal{P}$, let be $p_i = 1$, and $q_i = \min\{|A| : A \in \mathcal{A}^0 \text{ and } P_i \in A\}$. The following theorem holds.

Theorem 9. The above x_i satisfies condition (2). Moreover, the information dispersal algorithm Σ_1 obtained taking these values maximizes the information rate, that is $\varrho(\mathcal{A}, \Sigma_1) = \varrho_{\max}$.

Proof. Let be $A \in \mathcal{A}^0$. For all $P_i \in A$, there holds $q_i \leq |A|$. Hence,

$$\sum_{P_i \in A} \frac{p_i}{q_i} \geq \sum_{P_i \in A} \frac{1}{|A|} = |A| \cdot \frac{1}{|A|} = 1.$$

So each x_i satisfies equality (2).

To prove that the scheme Σ_1 reaches the bound on information rate, observe that each participant P_i receives m/q_i fragments, each of size $\log|F|/m$. Moreover, for all $i = 1, \dots, n$, $q_i = \min\{|A| : A \in \mathcal{A}^0 \text{ and } P_i \in A\} \geq \min\{|A| : A \in \mathcal{A}^0\} = \varrho_{\max}$. So, for $i = 1, \dots, n$ there holds

$$\log|G_i| = \frac{\log|F|}{m} \cdot \frac{m}{q_i} = \frac{\log|F|}{q_i} \leq \frac{\log|F|}{\varrho_{\max}}.$$

As $H(G_i) \leq \log|G_i|$ and $H(F) = \log(F)$, then

$$H(G_i) \leq \frac{H(F)}{\varrho_{\max}}.$$

Then $\max\{H(G_i) \ : \ 1 \leq i \leq n\} \leq H(F)/\varrho_{\max}$, so $\varrho(\mathcal{A}, \Sigma_1) \geq \varrho_{\max}$. From Theorem 5 it follows the equality. □

Second Technique

This second technique provide an information dispersal algorithm with maximal average information rate among the schemes with maximal information rate: this algorithm reaches the bound on information rate and gives to participants fragments no longer than necessary. It should be found solving the following linear programming problem called LP2.

$$\boxed{\begin{array}{lll} \text{Minimize} & M = \sum_{i=1}^{n} \beta_i & \text{subject to:} \\ & 0 \leq \beta_i \leq 1/\varrho_{\max}, & 1 \leq i \leq n \\ & \sum_{P_i \in A} \beta_i \geq 1, & \forall A \in \mathcal{A}^0 \end{array}}$$

Let be $M_{\min}^{(2)} = \sum_{i=1}^{n} \beta_i^*$ the solution to the linear programming problem LP2. As each β_i^*, for $i = 1, \dots, n$ is rational we can express it as a fraction $\beta_i^* = p_i/q_i$. The following theorem holds.

Theorem 10. The above β_i^* satisfies equation (2). Moreover, the information dispersal algorithm Σ_2 obtained taking $x_i = \beta_i^*$, for $i = 1, \dots, n$, maximizes the information rate, that is $\varrho(\mathcal{A}, \Sigma_2) = \varrho_{\max}$.

Proof. From definition of β_i^*, for all $A \in \mathcal{A}^0$, $\sum_{P_i \in A} \beta_i^* \geq 1$, so equality (2) is satisfied.

To prove that the scheme Σ_2 reaches the bound on information rate, observe that each participant P_i receives $m \cdot \beta_i^*$ fragments, each of size $\log|F|/m$. So, for $i = 1, \dots, n$ there holds

$$\log|G_i| = \frac{\log|F|}{m} \cdot m \cdot \beta_i^* = \log|F| \cdot \beta_i^* \leq \frac{\log|F|}{\varrho_{\max}}.$$

Then $\varrho(\mathcal{A}, \Sigma_2) \geq \varrho_{\max}$. From Theorem 5 it follows the equality. □

In this way, the sum of fragments distributed to participants is minimized, while all fragments are less than $\log|F|/\varrho_{\max}$.

3.5 How to Optimize the Average Information Rate

The linear optimization problem LP1 described in section 3.2 will be used to maximize the average information rate.
Let be $M_{\text{min}}^{(1)} = \sum_{i=1}^{n} \alpha_i^*$ the solution to the linear programming problem LP1. As each α_i^*, for $i = 1, \ldots, n$ is rational we can express it as a fraction $\alpha_i^* = p_i/q_i$. The following theorem holds.

Theorem 11. The above α_i^* satisfies equation (2). Moreover, the information dispersal algorithm Σ_3 obtained taking $x_i = \alpha_i^*$, for $i = 1, \ldots, n$, maximizes the average information rate, that is $\tilde{\varrho}(\mathcal{A}, \Sigma_3) = \frac{n}{M_{\text{min}}^{(1)}}$.

Proof. From definition of α_i^*, for all $A \in \mathcal{A}^0$, $\sum_{P_i \in A} \alpha_i^* \geq 1$, so equality (2) is satisfied.
To prove that the scheme Σ_3 reaches the bound on information rate, observe that each participant P_i receives $m \cdot \alpha_i^*$ fragments, each of size $\log |F|/m$. So, for $i = 1, \ldots, n$ there holds

$$\log |G_i| = \frac{\log |F|}{m} \cdot m \cdot \alpha_i^* \leq H(F) \cdot \alpha_i^*.$$

Then

$$H(G_i) \leq H(F) \cdot \alpha_i^*.$$

So, $\sum_{i=1}^{n} H(G_i) \leq H(F) \cdot \sum_{i=1}^{n} \alpha_i^* = H(F) \cdot M_{\text{min}}^{(1)}$. Hence,

$$\tilde{\varrho}(\mathcal{A}, \Sigma_3) = \frac{nH(F)}{\sum_{i=1}^{n} H(G_i)} \geq \frac{n}{M_{\text{min}}^{(1)}}.$$

From Theorem 7 it follows the equality. □

3.6 Comparison of the Techniques

Observe that the solution $M_{\text{min}}^{(2)}$ to the optimization problem LP2 is in general bigger than the solution $M_{\text{min}}^{(1)}$ to the problem LP1 in which the fragments may be bigger than $\log |F|/\varrho_{\text{max}}$. The following example shows that the three previous schemes don't give in general the same result.
Let $\mathcal{P} = \{P_1, \ldots, P_6\}$, and let be $\mathcal{A}^0 = \{\{P_1, P_2\}, \{P_2, P_3, P_4\}, \{P_2, P_5, P_6\}\}$ the basis of the access structure $\mathcal{A}$ on $\mathcal{P}$.
In the next table, we present for the three schemes Σ_1, Σ_2 and Σ_3 the following values in order to outline the differences between the techniques : the value $x_i = \frac{\log |G_i|}{\log |F|}$ for each $i = 1, \ldots, 6$, the value ϱ and the value $\tilde{\varrho}$.

	x_1	x_2	x_3	x_4	x_5	x_6	ϱ	$\tilde{\varrho}$
Σ_1	1/2	1/2	1/3	1/3	1/3	1/3	2	18/7
Σ_2	1/2	1/2	1/2	0	1/2	0	2	3
Σ_3	0	1	0	0	0	0	1	6

3.7 Modified Schemes

As we observe in section 3.1 the simple information dispersal algorithm $\Sigma(m,b)$ has the requirement $\log|F| = m\log q \geq m\log b$. So when the parameters m and b are big our algorithms work only for large files. Moreover, for special access structures m and b could be exponentially big in n.

To solve this problems, we propose now two information dispersal algorithms which work even for small files and which nearly reach the theoretical minimal bounds proved in section 3.2. The first algorithm is intended to maximize the information rate, while the second analizes the average information rate.

Algorithm Σ_4

This algorithm is simply an adaptation of algorithm Σ_1 to handle the case of small files. So, while for the algorithm Σ_1 there were for all $P_i \in \mathcal{P}$, $p_i = 1$ and $q_i = \min\{|A| \ : \ A \in \mathcal{A}^0 \text{ and } P_i \in A\}$, now, for the algorithm Σ_4 we take $p'_i = 1$ and $q'_i = \varrho_{\max} = \min\{|A| \ : \ A \in \mathcal{A}^0\}$. So, $m = \varrho_{\max}$, and $b = n$. Clearly $q'_i \leq q_i$, hence from theorem 9, for all $A \in \mathcal{A}^0$,

$$\sum_{P_i \in A} \frac{p'_i}{q'_i} \geq \sum_{P_i \in A} \frac{p_i}{q_i} \geq 1.$$

This proves that p'_i and q'_i satisfy inequality (2).
Each participant receives a fragment of size $\log|F|/\varrho_{\max}$. So, $\varrho(\mathcal{A}, \Sigma_4) \geq \varrho_{\max}$, and it follows that with this algorithm we obtain optimal information rate. Moreover, the constraint on the file size is $\log|F| \geq \varrho_{\max}\log n$, and $\varrho_{\max}$ satisfies $\varrho_{\max} \leq n$. So the algorithm Σ_4 is useful even for small files.

Algorithm Σ_5

This algorithm is an adaptation of algorithm Σ_3 which can be used with small files and which gives an average information rate close to optimum.
Suppose we have found the razios α^*_i and $M^{(1)}_{\min}$ solving the optimization problem LP1. Suppose $|F|^{1/n} > n(M^{(1)}_{\min} + 1)$. Let k be the biggest integer such that $|F|^{1/kn} > knM^{(1)}_{\min} + n$, and let be $b = \sum_{i=1}^{n} \lceil \alpha^*_i \cdot k \cdot n \rceil$. Use in the distribution scheme for Σ_5 an information dispersal algorithm with parameters $(k \cdot n, b)$. Observe that $b \leq knM^{(1)}_{\min} + n < |F|^{1/kn}$, so $\log|F| > kn\log b$. Then the simple algorithm of section 3.1 applies. We give to each participant P_i, $\lceil \alpha^*_i \cdot k \cdot n \rceil$ distinct fragments, each of size $\frac{\log|F|}{kn}$. So $\log|G_i| = \frac{\log|F|}{kn}\lceil \alpha^*_i \cdot k \cdot n \rceil$.
Now we give a bound on the average information rate of the algorithm.

$$\sum_{i=1}^{n} \log|G_i| \leq \sum_{i=1}^{n} \left(\frac{\log|F|}{kn}(\alpha^*_i \cdot k \cdot n + 1)\right) = (M^{(1)}_{\min} + \frac{1}{k})\log|F|.$$

Hence, from theorem 7

$$\frac{n}{M^{(1)}_{\min} + 1/k} \leq \tilde{\varrho}(\mathcal{A}, \Sigma_5) \leq \frac{n}{M^{(1)}_{\min}}.$$

It follows that with this algorithm we obtain an average information rate nearly optimal. Moreover, the constraint on the file size is $|F|^{1/n} > n\left(M_{\min}^{(1)} + 1\right)$, that is $\log|F| \geq n \log\left(n\left(M_{\min}^{(1)} + 1\right)\right)$, and $M_{\min}^{(1)}$ satisfies $M_{\min}^{(1)} \leq n$. So the algorithm Σ_5 is useful even for small files.

4 Computational Secret Sharing Schemes

In a computational secret sharing scheme, any qualified subset can reconstruct the secret but any non qualified subset obtain no computational information on the secret. The scheme presented here is a generalization of Krawczyk's one [9].

In this section we use the information dispersal algorithms described in the previous section to construct *short* secret sharing schemes for general access structures. We can choose $IDA \equiv \Sigma_i$ with $i \in \{1, \ldots, 5\}$, depending if we want optimize the information rate or the average information rate.

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be the set of participants and $\mathcal{A}$ be an access structure on $\mathcal{P}$. Let S be the space of secrets we want to share. We assume a secure (length preserving) private key encryption function with space of plaintext S, denoted ENC. Let be K the space of keys and F the space of ciphertexts of ENC. We now assume a perfect secret sharing scheme PSS for the access structure $\mathcal{A}$ and the set on secrets K and an information dispersal algorithm IDA for the same access structure and the set of files F.
For each participant $P_i \in \mathcal{P}$ we denote by G_i the set of possible fragments given to participant P_i with IDA, and by V_i the set of possible shares given to participant P_i with PSS. Moreover, we denote by $W_i = G_i \times V_i$ the set of possible shares given to participant P_i in the scheme SS which we are going to describe. We consider uniform probability distributions both over S and over K.

Distribution Scheme of SS:

- Chose a random encryption key $k \in K$. Encrypt the secret $s \in S$ using the encryption function ENC under the key k, let $f = ENC_k(s)$.
- Using IDA partition the encrypted file f into n fragments $g_1, \ldots, g_n$, and distribute them to the participants in $\mathcal{A}$.
- Using PSS generate n shares for the key k, denoted $v_1, \ldots, v_n$, and distribute them to the participants in $\mathcal{A}$.

The share of each participant P_i, $i = 1, \ldots, n$ consists on $w_i = (g_i, v_i)$.

Reconstruction Scheme of SS:

- Each set of participants A in the access structure collect their shares.
- Using IDA reconstruct f out of the collected values g_i for all $P_i \in A$.
- Using PSS recover the key k out of v_i for all $P_i \in A$.
- Decrypt f using k to recover the secret s.

The next theorem is similar to Krawczyk's one [9].

Theorem 12. The above scheme SS constitutes a computationally secure secret sharing scheme for the access structure $\mathcal{A}$ provided that ENC is a secure encryption function and PSS a perfect secret sharing scheme.

Proof. The feasibility for a set of participants A to reconstruct the encrypted secret f out of the fragments is inherited from the properties of the algorithm IDA. Also the reconstruction of the key k out of v_i for all $P_i \in A$ is guaranteed by the secret sharing scheme PSS. Knowledge of f and k permits deriving s using the decryption function to each set of participants in the access structure $\mathcal{A}$.

As for the secrecy against a coalition of participants B not belonging to the access structure, the intuitive idea is the following. The fragments corresponding to f of all participants in B give no more information on s than f itself. On the other hand, the fragments corresponding to k of all participants in B give no information at all on k. Therefore participants in B cannot learn something about s. □

The Size of Shares

The length of each share for $i = 1, \ldots, n$, is $\log|W_i| = \log|G_i| + \log|V_i|$. So, the length of the shares depends both on the information dispersal algorithm and on the perfect secret sharing scheme used to construct the scheme. Depends on what information dispersal algorithm we choose, the size of G_i is minimal for the corresponding definition. And the size of V_i does not depend on the secret size but only on the security parameter.

However, observe that for general access structures the size $\log|V_i|$ of shares of perfect secret sharing schemes used in order to share the enciphering key k should be exponentially large respect to the size of the secret key $\log|K|$: an upper bound better than exponential is not known for the length of shares in the general case. Moreover, Csirmaz [5] proved that there are access structures on n elements so that any perfect secret sharing scheme must assign a share which is of size at least $\frac{n}{\log n}$ times the size of the secret k.

We have better upper bounds when the access structure is based on graphs. If, for example, the graph on which the access structure is based is complete multipartite, then there exists an ideal perfect secret sharing scheme for $\mathcal{A}$ (see [3]) and the size of the shares becomes $\log|S|/\varrho_{\max} + \log|K|$. Otherwise, using bounds found in [17] we can say that $\log|V_i| \leq \log|K|(\Delta + 1)/2$, where Δ is the maximum degree of the graph. Moreover, better bounds on V_i can be obtained if the graph is acyclic.

5 Conclusions

We have shown how to realize computational secret sharing schemes for general access structures. Our schemes are nearly optimal: the size of each share is equal to the minimal theoretical bound plus a piece of information whose length does not depend on the secret size but just on the security parameter.

We remark that the size of the shares that must be kept secret is an important issue: in many cases such shares must be kept in mind or in tamper-resistant devices, so they must be very small. In our scheme, only V_i must be secret, the piece G_i could be in a hard disk or in a floppy disk. Moreover, the size of the V_i does not depend on the size of the secret file. Hence for very long files our schemes are very useful.

Finally, observe that computational schemes are not weaker than perfect ones in practical viewpoint since most of the time people uses an encryption function to distribute the shares or a pseudo-random generator to produce them. Hence our schemes are very convenient and practical for very long file.

References

1. J. Benaloh, and J. Leichter, *Generalized secret sharing and monotone functions*, in "Advances in Cryptology - CRYPTO '88", S. Goldwasser Ed., " Lecture Notes in Computer Science", Vol. 403, Springer-Verlag, Berlin, 1988, pp. 27–35.
2. C. Blundo, A. De Santis, L. Gargano, and U. Vaccaro, *On the Information Rate of Secret Sharing Schemes*, in "Advances in Cryptology - CRYPTO 92", Ed. E. Brickell, "Lecture Notes in Computer Science", Vol. 740, E. Brickell Ed., Springer-Verlag, pp. 149–169, 1993.
3. E. F. Brickell and D. M. Davenport, *On the Classification of Ideal Secret Sharing Schemes,* J. Cryptology, Vol. 4, No. 2, pp. 123–124, 1991.
4. R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro, *On the Size of Shares for Secret Sharing Schemes,* Journal of Cryptology, Vol. 6, No. 3, pp. 157–169, 1993.
5. L. Csirmaz, *Size of Shares Must Be Large*, in "Advances in Cryptology – Eurocrypt '94", Lecture Notes in Computer Science, A. De Santis Ed., Springer-Verlag.
6. E. D. Karnin, J. W. Greene, and M. E. Hellman, *On Secret Sharing Systems,* IEEE Trans. on Inform. Theory, Vol. IT-29, No. 1, pp. 35–41, Jan. 1983.
7. I. Csiszar and J. Körner, *Information Theory. Coding Theorems for Discrete Memoryless Systems,* Academic Press, 1981.
8. R. G. Gallager, *Information Theory and Reliable Communications*, John Wiley & Sons, New York, NY, 1968.
9. H. Krawczyk, *Secret Sharing Made Short,* in "Advances in Cryptology - CRYPTO '93", D. Stinson Ed., " Lecture Notes in Computer Science", Vol. 773, Springer-Verlag, Berlin, 1994.
10. K. Kurosawa, W. Ogata, S. Tsujii, *Nonperfect secret sharing schemes,* in "Advances in Cryptology - AUSCRYPT '92".
11. K. Kurosawa, W. Ogata, K. Okada, K. Sakano, S. Tsujii, *Nonperfect secret sharing schemes and Matroids,* in "Advances in Cryptology - EUROCRYPT '93".
12. M. Naor, and R. M. Roth, Optimal File Sharing in Distributed Networks, Proceedings of 32nd IEEE Symposium on Foundations of Computer Science, 1991, pp. 515–525.
13. C. H. Papadimitriou, and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity,* Prentice Hall, 1982.

14. M. O.Rabin, *Efficient Dispersal of Information for Security, Load Balancing and Fault Tolerance*, Journal of ACM, Vol. 36, No. 2, 1989, pp. 335–348.

15. M. O. Rabin, *The Information Dispersal Algorithm and its Applications*, in "Sequences: Combinatorics, Compression, Security and Transmission", R. M. Capocelli Ed., Springer-Verlag, 1990, pp. 406–419.

16. A. Shamir, *How to Share a Secret* Communications of the ACM, Vol. 22, n. 11, pp. 612–613, Nov. 1979.

17. D. R. Stinson, *Decomposition Constructions for Secret Sharing Schemes*, IEEE Trans. on Inform. Theory, Vol. IT-40, pp. 118–125, 1994.

A Information theory

In this appendix we review the information theoretic concepts we are going to use. For a complete treatment of the subject the reader is advised to consult [7] and [8].

Given a probability distribution $\{p(x)\}_{x \in X}$ on a set X, we define the *entropy* of X, $H(X)$, as

$$H(X) = -\sum_{x \in X} p(x) \log p(x)^3.$$

The entropy $H(X)$ is a measure of the average information content of the elements in X or, equivalently, a measure of the average uncertainty one has about which element of the set X has been chosen when the choices of the elements from X are made according to the probability distribution $\{p(x)\}_{x \in X}$. The entropy enjoys the following property

$$0 \leq H(X) \leq \log |X|, \tag{3}$$

where $H(X) = 0$ if and only if there exists $x_0 \in X$ such that $p(x_0) = 1$; $H(X) = \log |X|$ if and only if $p(x) = 1/|X|$, $\forall x \in X$.

Given two sets X and Y and a joint probability distribution $\{p(x,y)\}_{x \in X, y \in Y}$ on their Cartesian product, the *conditional entropy* $H(X|Y)$, also called the equivocation of X given Y, is defined as

$$H(X|Y) = -\sum_{y \in Y} \sum_{x \in X} p(y)p(x|y) \log p(x|y).$$

The conditional entropy can be written as $H(X|Y) = \sum_{y \in Y} p(y)H(X|Y = y)$ where $H(X|Y = y) = -\sum_{x \in X} p(x|y) \log p(x|y)$ can be interpreted as the average uncertainty one has about which element of X has been chosen when the choices are made according to the probability distribution $\{p(x|y)\}_{x \in X}$, that is, when it is known that the value chosen from the set Y is y. From the definition of conditional entropy it is easy to see that

$$H(X|Y) \geq 0. \tag{4}$$

[3] All logarithms in this paper are of base 2

If we have $n+1$ sets $X_1, \ldots, X_n, Y$ the entropy of $X_1 \ldots X_n$ given Y can be written as

$$H(X_1 \ldots X_n|Y) = H(X_1|Y) + H(X_2|X_1Y) + \cdots + H(X_n|X_1 \ldots X_{n-1}Y) \quad (5)$$

The *mutual information* between X and Y is defined by

$$I(X;Y) = H(X) - H(X|Y) \quad (6)$$

and enjoys the following properties:

$$I(X;Y) = I(Y;X), \quad (7)$$

and

$$I(X;Y) \geq 0,$$

from which one gets

$$H(X) \geq H(X|Y), \quad (8)$$

with equality if and only if X and Y are independent.