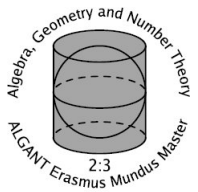
Alfonso Cevallos Manzano

Reducing the Share Size in Robust Secret Sharing

Master's Thesis, defended on October 18, 2011

Thesis Advisors:
Ronald Cramer (CWI & UL)
Serge Fehr (CWI)



Mathematisch Instituut
Universiteit Leiden

To Sarita

Contents

1 Introduction

1.1 The Idea of Secret Sharing

A secret sharing (SS) scheme is a method used to distribute a secret (for instance, a bit-string), among a determined group of players, each of whom receives a share of the secret. The goal of the method is to allow any sufficiently large subset of the players to reconstruct the secret, while leaking no knowledge thereof to individuals or small groups of players. This concept has proved very useful both practically, as means to keep important information both from overexposure and from possible loss, as well as theoretically, as a basis to general multi-party secure protocols.

As examples of practical uses, a bank can program its safe to open only when at least three coworkers enter their passwords; or a computational system may split a file into shares, that are stored in hard drives physically distant one from the other, so that it is highly unlikely for a thief to be able to seize more than one of the shares, but a single share lost by the system to the thief neither reveals any useful information to the latter nor prevents the recovery of the file by the former.

A SS scheme is robust if it offers protection not only against loss but also against modification of some of the shares, and still guarantees a correct reconstruction with high probability. We focus on schemes tolerating a maximum amount of modified shares (we study what this maximum value is), and with a failure probability that is exponentially small over a security parameter. Naturally, the size of each share grows as a function of the number of players and the security parameter.

The concept of robust secret sharing (RSS) is closely related to that of verifiable secret sharing (VSS). While RSS tolerates only corrupted players, VSS in addition tolerates a corrupted dealer (the distributor of the shares), and guarantees with a small error probability that the shares actually reconstruct a unique secret. Notice that VSS is not in the scope of this paper, and we require the dealer to be honest.

1.2 Our Contribution

We analyse RSS schemes that are information-theoretically secure. We prove that any scheme with zero probability of failure requires more than $2/3$ of the players to be honest, while a scheme with an exponentially small probability of failure requires only a majority of honest players. We also prove that the average bit-length of a share is lower-bounded by the bit-length of the secret.

We introduce our new scheme in section 5, and compare its performance to those of other common schemes. We consider for all schemes one same scenario where an m-bit secret is shared among $n = 2t + 1$ players, $t + 1$ of whom are honest, and the failure probability is required to be at most $2^{-\lambda}$, and obtain the following results (the logarithmic terms will be ommited): The scheme by Rabin and Ben-Or [13] is efficient (its algorithms run on polynomial time over n) and its share size is $m + O(n \cdot \lambda)$. On the other hand, the scheme by Cramer, Fehr and Damgård [5] is not efficient, but it offers an improved share size of $m + O(n + \lambda)$. The new scheme achieves simultaneously efficiency and a share size of $m + O(n + \lambda)$.

2 Preliminaries

2.1 Notation

$\mathbb{R}^+$ is the set of positive real numbers, $\mathbb{N}$ is the set of positive integers (thus 0 is excluded), and $[n]$ is the set of the first n positive integers. Given a set A, a positive integer n, a vector $V = (v_1, \cdots, v_n) \in A^n$ and an index subset $I \subset [n]$, we represent by V_I the sub-vector $(v_i)_{i \in I} \in A^{|I|}$; and for any element $v \in A$, we denote by v^n the vector in A^n where each coordinate equals v.

We refer by $\log(\cdot)$ to the logarithm function with base 2, and by $\ln(\cdot)$ to the natural logarithm function. If K is a field, and t is a non-negative integer, we denote by $K[X]_{\leq t}$ the set of all polynomials with coefficients in K and degree at most t.

2.2 Probabilistic Algorithms

The algorithms considered in this paper are always-halting, and may be deterministic or probabilistic. A probabilistic algorithm has access to one or more black boxes, each of which, when called upon, returns a *truly* random value chosen uniformly from a specified finite domain. By definition, it is impossible to gain any partial information about the next value a black box will return. A *run* of a probabilistic algorithm is specified by its input plus the values returned by all its black boxes; and, given a specific input, we can view the output of the algorithm as a random variable. Finally, the definition of the algorithms in a scheme may be public and available to any adversary, as we base the security of the scheme solely on the randomness of the black boxes outputs.

2.3 Description of Model

In this paper, we consider information-theoretic security, i.e. security against an adversary with unbounded computational capabilities. The dealer (the person who holds a secret and computes and distributes the shares) and the reconstructor (the person who collects the shares and computes a secret from them) are not required to be share-holding players. They can be anybody, but both must be honest. We focus on threshold schemes (i.e. any given subset of players either knows everything or nothing about the secret, depending only on the size of the subset). The adversary is free to corrupt any subset of players of a specific size, and the secret chosen by the dealer to be shared may have any probability distribution.

3 Secret Sharing

3.1 Definitions

Definition 3.1. A secret sharing (SS) scheme Ω is a tuple (n, A, B, SH, RE) consisting of:

- A positive integer n, called *the number of players*;
- A finite set A with $|A| \geq 2$, whose elements are called *secrets*;
- A finite set B, whose elements are called *shares*;
- An algorithm SH (usually probabilistic), called the *sharing algorithm*, that takes as input a secret $s \in A$, and outputs a vector of n shares $(s_1, \cdots, s_n) \in B^n$; and
- An algorithm RE, called the *reconstruction algorithm*, that takes as input a vector $(s'_1, \cdots, s'_n) \in (B \cup \{\perp\})^n$, and outputs either a secret $s' \in A$ or $\perp$. Here, $\perp$ is a fixed symbol, not contained in $A \cup B$, that represents a *missing share* in the input, and *failure to reconstruct the secret* in the output.

We consider a model where there are n enumerated players $P_1, \cdots, P_n$, and a SS scheme $\Omega = (n, A, B, SH, RE)$ publicly known to them. The model begins with a *sharing phase*, where a *dealer* D arbitrarily picks a secret $s \in A$ and distributes it among the n players: each player P_i receives the share s_i outputted by SH from input s, while s itself remains unknown to them. At a later *reconstruction phase*, a *reconstructor* R collects the shares from the players and tries to retrieve the secret: he gives the collected shares as input to RE, and hopes to receive the secret s as output. The behaviours of the players during the reconstruction phase, and the corresponding results outputted by RE, are to be analysed in the remainder of the paper.

Definition 3.2. Consider a SS scheme $\Omega = (n, A, B, SH, RE)$. For a given secret $s \in A$, an index subset $I \subset [n]$, and a share sub-vector $W \in B^{|I|}$, we define the *matching probability* $P^M(s, I, W)$ as the probability that the sharing algorithm outputs a vector $SH(s) = V$ such that $V_I = W$; i.e. $P^M(s, I, W) = P\left[(SH(s))_I = W\right]$.

Since the players know how the sharing algorithm is defined, they can calculate the matching probability function and use it to see what their shares reveal about the secret. For instance, a group of players may compute what secrets have the highest matching probability corresponding to their shares. The following properties of a SS scheme tell just how revealing its matching probability is.

Definition 3.3. Consider a SS scheme $\Omega = (n, A, B, SH, RE)$ and integers t, r satisfying $0 \leq t < r \leq n$.

Ω is *t-private* if, for any index subset $I \subset [n]$ of size $|I| \leq t$, and any share sub-vector $W \in B^{|I|}$, the matching probability $P^M(s, I, W)$ does not depend on the secret $s \in A$.

Ω is *r-reconstructible* if, for any index subset $I \subset [n]$ of size $|I| \geq r$, any secret $s \in A$, and any run of SH ouputting shares $SH(s) = V$, we obtain $RE(V') = s$, where we define $V' \in (B \cup \{\perp\})^n$ by $V'_I = V_I$ and $V'_{[n]\setminus I} = \perp^{n-|I|}$.

In other words, if Ω is t-private, for any subset of at most t players the distribution probability of their shares is independent of the secret, and thus their shares give zero information about the secret that generated them. And if Ω is r-reconstructible, the shares of any r or more players automatically identify the secret.

Definition 3.4. Consider a SS scheme $\Omega = (n, A, B, SH, RE)$.

For every index $i \in [n]$, the *i-th share range* B_i of Ω is the subset of B consisting of all possible values of the i-th coordinate of the outputs of SH, over all possible inputs $s \in A$ and all possible runs of SH.

The *secret size* and the *share size* of Ω are respectively $l_{secret} = \log |A|$ and $l_{share} = \frac{1}{n} \sum_{i=1}^{n} \log |B_i|$.

The i-th share range is the set of all shares that the i-th player may receive, and the secret size and share size of Ω are the average of the number of bits required to represent, respectively, a secret or a share in a computational system.

A scheme with the properties of privacy and reconstructibility splits a secret into several parts, and simultaneously grants access to large groups of players and blocks access to small groups. In order to maximize this dual power of the scheme, we would like to eliminate the middle ground, i.e. the situation where a group receives only partial information about the secret. This maximization however comes with a cost on the share size.

Definition 3.5. Let Ω be a SS scheme with n players. If there is an integer $0 \leq t < n$ such that Ω is t-private and $(t+1)$-reconstructible, Ω is called *threshold*, or *t-threshold*.

Proposition 3.6. *If the SS scheme $\Omega = (n, A, B, SH, RE)$ is threshold, then $|B_i| \geq |A|$ for all $i \in [n]$; and in particular, $|B| \geq |A|$, and $l_{share} \geq l_{secret}$.*

Proof. Suppose that a run of SH, for an unknown secret $s \in A$, outputs the shares $SH(s) = (s_i)_{i \in [n]}$. Let t be the number for which Ω is t-threshold, fix an index $i \in [n]$, and fix any subset of t players all distinct from P_i. By t-privacy, their t shares give zero information about the secret s (so they deem all elements of A possible), but by $(t+1)$-reconstructibility, knowledge of the share $s_i \in B_i$ would uniquely determine the secret to them. This defines a surjective function from B_i to A (or to $A \cup \{\perp\}$), and thus $|B_i| \geq |A|$. □

Definition 3.7. A threshold SS scheme $\Omega = (n, A, B, SH, RE)$ is *ideal* if $|A| = |B|$ (and consequently $l_{share} = l_{secret}$).

Next, we give two examples of ideal threshold SS schemes that are trivial (in the sense that they are extreme cases).

Scheme 3.8. For any positive integer n and any finite set A with $|A| \geq 2$, we define the ideal 0-threshold SS scheme $\Omega = (n, A, A, SH, RE)$ as:

SH For input $s \in A$, output s^n.

RE For input $(s'_i)_{i \in [n]}$, if some s'_i is not $\perp$ then output s'_i, otherwise output $\perp$.

Scheme 3.9. For any positive integer n and any finite non-trivial group $(G, *)$, we define the ideal $(n-1)$-threshold SS scheme $\Omega = (n, G, G, SH, RE)$ as:

SH For input $s \in G$, output $(s_i)_{i\in[n]}$, where the shares s_i are picked at random from G with the condition that $s_1 * \cdots * s_n = s$.

RE For input $(s'_i)_{i\in[n]}$, if some $s'_i = \perp$ then output $\perp$, otherwise output $s'_1 * \cdots * s'_n$.

Notice that the previous examples are not exactly SS *schemes*, but rather *families of schemes*, because they define a scheme for each value of one or more parameters. Working with families of schemes permits to analyse the efficiency of its algorithms. In this paper we equate the notion of efficiency with that of *polynomial time complexity*.

Definition 3.10. Given a family of SS schemes indexed by a set C, $\{\Omega(x)\}_{x\in C}$, and a function $\psi : C \to \mathbb{R}^+$, we say that the family is *poly-time in* $\psi(x)$ if both the sharing algorithms and the reconstruction algorithms of the family have running times upper-bounded by a polynomial expression in $\psi(x)$.

For instance, if $\mathbb{G}$ is any set of finite non-trivial groups, then we can use the definition of scheme 3.9 to describe a family $\Omega(n, G) = (n, G, G, SH, RE)$ indexed by the set $\mathbb{N} \times \mathbb{G}$; and if besides $\mathbb{G}$ consists only of groups G whose operations (including sampling random elements) can be performed in polynomial time in $\log |G|$, it can be proved that this family is poly-time in $n \log |G| = n \cdot l_{secret}$.

When a family $\Omega(x)$ is given, and the index set is clear from the context, for brevity we may refer to the whole family as a scheme, and say for instance that the scheme is poly-time in some function $\psi(x)$.

3.2 Shamir Scheme

Secret Sharing was invented by both Adi Shamir [16] and George Blakley [2] independently in 1979. In what follows, we describe a non-trivial ideal threshold scheme invented by Shamir, that relies on the Lagrange interpolation theorem.

Theorem 3.11 (Lagrange Interpolation Theorem)**.** *Let K be a field and t be a non-negative integer, and let $x_0, ..., x_t \in K$ be $t + 1$ distinct points on K. Then the map*

$$\begin{aligned} \phi : K[X]_{\leq t} &\to K^{t+1} \\ f &\mapsto (f(x_0), ..., f(x_t)) \end{aligned}$$

is a bijection, and its inverse is

$$\phi^{-1}(y_0, ..., y_t) = f(X) := \sum_{j=0}^{t} y_j \prod_{k\neq j} \frac{X - x_k}{x_j - x_k}.$$

Scheme 3.12 (Shamir)**.** For integers n, t, and a finite field K, satisfying $0 \leq t < n < |K|$, we define the Shamir SS scheme $\Omega^S(t, n, K) = (n, K, K, SH, RE)$ in the following way. Fix $n+1$ distinct points $x_0, x_1, \cdots, x_n$ in the field K (this is possible because $n < |K|$). Then define the algorithms:

SH For input $s \in K$, pick a random polynomial $f(X) \in K[X]_{\leq t}$ with the condition that $f(x_0) = s$, and output $(f(x_1), \cdots, f(x_n))$.

RE For input $(s'_i)_{i\in[n]}$, find an index subset $I \subset [n]$ of size $|I| = t+1$ such that $s'_i \neq \perp$ for all $i \in I$, or output $\perp$ if such a subset cannot be found. Then, compute the polynomial $g(X) \in K[X]_{\leq t}$ for which $g(x_i) = s'_i$ for all $i \in I$ (existence and uniqueness of such polynomial are guaranteed by Theorem 3.11). Finally, output $g(x_0)$.

Proposition 3.13. *The Shamir SS scheme $\Omega^S(t, n, K) = (n, K, K, SH, RE)$ is t-threshold, ideal and poly-time in $n \log |K| = n \cdot l_{secret}$.*

Proof. The $(t + 1)$-reconstructibility and the poly-time properties are evident from the definition of the scheme, from Theorem 3.11, and from the fact that the arithmetic in any finite field K can be performed in polynomial time in $\log |K|$. Also, if the scheme is threshold, the ideality property becomes aparent from definition. Thus it only remains to prove t-privacy.

Fix a secret $s \in K$, a number $0 \leq p \leq t$, an index subset $I \in [n]$ of size $|I| = p$, and a share sub-vector $W \in K^p$. The sharing algorithm SH, for input s, can have $|K|^t$ distinct runs (one for each polynomial $g(X) \in K[X]_{\leq t}$ s.t. $g(x_0) = s$), all equally probable, and only $|K|^{t-p}$ of them output a vector V such that $V_I = W$ (again by Thm. 3.11). Therefore the matching probability, $P^M(s, I, W) = \frac{|K|^{t-p}}{|K|^t} = |K|^{-p}$, is independent of s. This completes the proof. □

It is also worth mentioning another special property of the Shamir scheme.

Definition 3.14. A SS scheme $\Omega = (n, A, B, SH, RE)$ is *linear* if A and B are abelian groups, and for any two secrets $s, s' \in A$, the random variables $SH(s)+SH(s')$ and $SH(s+s')$ have the same probability distribution over B^n.

Proposition 3.15. *Shamir SS scheme $\Omega^S(t, n, K)$ is linear.*

The proof is direct from the definition and is not included.

The linearity property implies that for any two secrets s and s', we always obtain $RE\,(SH(s) + SH(s')) = s + s'$, and in general any linear transformation on the secret produces the same transformation on the shares, and viceversa.

4 Robust Secret Sharing

4.1 Definitions

A secret sharing scheme guarantees privacy against a small group of players, and correct reconstruction against erasure of a small number of shares. Now we expand our model to consider the possibility that some of the shares are modified. This is, of course, a much greater attack, and for that reason we will consider schemes admitting a small failure probability in the reconstruction.

Let's define the possible behaviours of a player. An *honest player* receives her share in the sharing phase, keeps it completely private and safe against loss or modification, and delivers it to the reconstructor in the reconstruction phase. A *corrupted player*, on the other hand, may communicate his share to other corrupted players, and may pass a bogus share to the reconstructor. For a t-threshold scheme, we will only study situations with up to t corrupted players.

In many applications of secret sharing, a player may become corrupted by accident, such as channel noise or a system failure. However, for greater generality, we suppose that there is an external player, called the *adversary*, who plots an attack against a t-threshold scheme, with the intention to gain information over the secret and prevent the correct recovery of it by the reconstructor. He does so following some strategy, and has the ability to choose t players (who will become corrupted), learn their shares, and modify or erase each of them at will. No assumption about the computational capability of the adversary is made.

Remark 4.1. Because of the simplified definitions in this paper, we require that the adversary not have access to the communication channels between the honest players and the reconstructor. Notice nonetheless that the Rabin Ben-Or scheme (4.9) and our new scheme (5.1) also work for a model where the communication channels between the players and the reconstructor are public, but the delivery of the shares is made by stages, in a more interactive process.

Definition 4.2. Given a t-threshold SS scheme $\Omega = (n, A, B, SH, RE)$, we define an *adversary strategy* for Ω as a pair (I, MO) consisting of an index subset $I \subset [n]$ of size $|I| = t$, and a (possibly probabilistic) *modification algorithm* MO that takes as input a share sub-vector $W \in B^t$, and outputs a sub-vector $W' \in (B \cup \{\perp\})^t$.

Definition 4.3. For a scheme Ω and an adversary strategy (I, MO) as above, and a secret $s \in A$, we define the *failure probability* $P^F(s, I, MO)$ as the probability that $RE(V') \neq s$, where V' is the modified share vector defined by $V'_I = MO(V_I)$, $V'_{[n]\setminus I} = V_{[n]\setminus I}$, and $SH(s) = V$.

The failure probability $P^F(s, I, MO)$ is exactly the probability that the adversary manages to prevent the correct reconstruction of the original secret s, when his strategy consists in choosing the t players signaled by I and modifying their shares with algorithm MO.

Definition 4.4. Consider a t-threshold SS scheme Ω.

If there is a real number $\lambda > 1$ such that $P^F(s, I, MO) \leq 2^{-\lambda}$ for any adversary strategy (I, MO) and any secret s, Ω is called *robust*, or *λ-robust*.

If $P^F(s, I, MO) = 0$ for any adversary strategy (I, MO) and any secret s, Ω is called *perfectly robust*.

Notice that by definition of a robust SS (RSS) scheme, the failure probability must always be strictly smaller than 1/2.

4.2 Existence of RSS Schemes

In the following three propositions we study the existence of robust t-threshold schemes, depending on the values of t.

Proposition 4.5. *A t-threshold SS scheme $\Omega = (n, A, B, SH, RE)$ with $t \geq \frac{n}{2}$ cannot be robust.*

Proof. Consider such a scheme Ω, and consider an adversary strategy (I, MO) where $I = [t]$ and MO outputs the vector $\perp^t$ regardless of the input (i.e. the first t shares are erased). Since $n - t \leq t$ and by t-privacy, the modified share vector has the same probability distribution for all secrets, so RE will have to output either $\perp$ or a blind guess. Therefore:

$$\sum_{s \in A} P[RE(V') = s] = \sum_{s \in A} \big(1 - P^F(s, I, MO)\big) \leq 1,$$

where V' is the modified share vector coming from s and (I, MO).

Thus there must be a secret $s \in A$ for which $1 - P^F(s, I, MO) \leq \frac{1}{|A|} \leq \frac{1}{2}$ (remember that $|A| \geq 2$ by definition of a SS scheme), so $P^F(s, I, MO) \geq \frac{1}{2}$, and Ω is not robust. □

Proposition 4.6. *A t-threshold SS scheme $\Omega = (n, A, B, SH, RE)$ with $t \geq \frac{n}{3}$ cannot be perfectly robust.*

Proof. Consider such a scheme Ω. Fix two distinct secrets $s, s' \in A$ and a partition of the index set $[n] = I \cup I' \cup J$ so that $|I| = |I'| = t$ and $|J| = n - 2t$. Since $n - 2t \leq t$ and by t-privacy, for both secrets the probability distribution of the output shares indexed by J is the same, so there are possible outputs $SH(s) = V$ and $SH(s') = V'$ such that $V_J = V'_J$.

Now, define the vector $\overline{V} \in B^n$ by $\overline{V}_{[n]\setminus I} = V_{[n]\setminus I}$ and $\overline{V}_{[n]\setminus I'} = V'_{[n]\setminus I'}$. This way, $\overline{V}$ differs both from V and from V' in at most t coordinates, so there are adversary strategies that modify V and V', respectively, to $\overline{V}$. Therefore $\overline{V}$ can come from two different ⟨secret, adversary strategy⟩ pairs. Since $RE(\overline{V})$ is either $\neq s$ or $\neq s'$, the failure probability for one of these pairs is non-zero, and the scheme is not perfectly robust. □

Proposition 4.7. *For $t < n/3$, Shamir SS scheme $\Omega^S(t, n, K)$ becomes perfectly robust under this new definition of the reconstruction algorithm:*

RE *Let $x_0, x_1, \cdots, x_n \in K$ be the $n+1$ points defined along with the scheme, and consider the Reed-Solomon code $RS_{K,\{x_1,\cdots,x_n\}}[n, t+1] = (C, D)$ (see definition A.1). For input share vector $V \in (K \cup \{\perp\})^n$, replace any coordinate equal to $\perp$ by any value in the field to obtain $V' = (s'_i)_{i \in [n]} \in K^n$. Then, let $g(X) = D(V') \in K[X]_{\leq t}$ (which is the polynomial satisfying $g(x_i) = s'_i$ for a maximum number of indices i), and output $g(0)$.*

Proof. Suppose that originally, in the sharing phase, SH received a secret $s \in K$ as input and picked a polynomial $f(X) \in K[X]_{\leq t}$ such that $f(x_0) = s$, in

order to produce the shares. Notice then that the share vector will be exactly $SH(s) = C(f(X))$. For any adversary strategy, the input V of RE, and also V', will have at most t modified shares, but the hypothesis $t < n/3$ implies that $t \leq \left\lfloor \frac{n-t-1}{2} \right\rfloor$. Thus, by Proposition A.2, $D(V') = f(X)$, and $f(x_0) = s$ is the correct secret. □

Remark 4.8. The Reed-Solomon code can be applied efficiently. Therefore, for integers t, n and a finite field K satisfying $0 \leq 3t < n < |K|$, there is a t-threshold SS scheme $\Omega^S = (n, K, K, SH, RE)$ that is perfectly robust, ideal, linear and poly-time in $n \log |K|$ (by Propositions 3.13, 3.15, 4.7 and A.3).

In conclusion, if a (t, n)-scheme represents a t-threshold SS scheme with n players:

1. For $n \leq 3t$, there is no perfectly robust (t, n)-scheme;
2. For $n \geq 3t+1$, there are examples of perfectly robust (t, n)-schemes (with very satisfactory properties);
3. For $n \leq 2t$, there is no robust (t, n)-scheme;
4. For $n \geq 2t+1$, there are examples of λ-robust (t, n)-schemes for any given $\lambda > 1$.

We have proved the first three points, and the last one is proved by the examples that follow in this section. We will focus exclusively on the limiting case $n = 2t + 1$, for schemes with the greatest tolerance to attacks.

Robustness is achieved in general by adding redundant information to the shares, and thus increasing the share size from the ideal situation $l_{share} = l_{secret}$. For one same level of robustness, we assess different schemes by comparing their share sizes and their efficiency.

4.3 Rabin Ben-Or Scheme

We will present the RSS scheme invented by Tal Rabin and Michael Ben-Or in 1989 [13]. It bases its robustness on message authentication codes (MAC, see Appendix B). In this paper we a use a different, improved version of the original MAC employed by Rabin and Ben-Or.

In the sharing phase the dealer generates keys and tags that allow every player to verify the authenticity of the share of every other player. These keys and tags are defined over a finite field whose size is chosen big enough so that the MAC has the needed error probability.

In the reconstruction phase, the reconstructor recovers a secret from the point of view of each player (using the shares that this player accepts), and outputs the majority value of them.

Scheme 4.9 (Rabin Ben-Or)**.** For a positive odd integer $n = 2t+1$, a finite field K satisfying $|K| > n$, and a real number $\lambda > 1$, we define the Rabin Ben-Or SS scheme $\Omega^{RB}(n, K, \lambda) = (n, K, K \times F^{3n}, SH, RE)$, in the following way:

Define the integers $q = \lceil \lambda + 2 \log n + \log \log |K| - 2 \rceil$ and $d = \left\lceil \frac{\log |K|}{q} \right\rceil$, let F be the field of order 2^q, and fix and injective function ψ from K to F^d. Define the algorithms:

SH For input secret $s \in K$, use Shamir scheme $\Omega^S(t, n, K)$ (scheme 3.12) to generate the vector $(s_i)_{i\in[n]} \in K^n$. Then, for every pair of indices $i, j \in [n]$, pick a random key $\kappa_{ij} \in F^2$, and use the polynomial-evaluation MAC (definition B.3) to define the tag $\tau_{ji} = h_{F,d}(\psi(s_j), \kappa_{ij}) \in F$. Output $(s_i, \kappa_{i1}, \cdots, \kappa_{in}, \tau_{i1}, \cdots, \tau_{in})_{i\in[n]}$.

RE Let the input shares be $(s'_i, \kappa'_{i1}, \cdots, \kappa'_{in}, \tau'_{i1}, \cdots, \tau'_{in})_{i\in[n]}$. For every index $i \in [n]$, do the following:

Reconstruction according to i: Let $I \subset [n]$ be the set of indices j for which the equality $\tau'_{ji} = h_{F,d}(\psi(s'_j), \kappa'_{ij})$ holds. Define the vector $W \in K^n$ as $W_I = (s'_j)_{j\in I}$, $W_{[n]\setminus I} = \perp^{n-|I|}$, and use Shamir scheme to reconstruct S_i from the share vector W.

If a majority of coordinates in the vector $(S_i)_{i\in[n]}$ have one same value s', output s'; otherwise, output $\perp$.

Proposition 4.10. *The scheme $\Omega^{RB}(n, K, \lambda)$ is poly-time in $n\lambda l_{secret}$, and its share size is $l_{share} \le l_{secret} + 3n\lambda + n \cdot O(\log n + \log l_{secret})$.*

Proof. The core part of both the sharing and the reconstruction algorithms is to calculate n^2 times a polynomial-evaluation MAC, and by prop. B.5 this MAC admits an algorithm that is poly-time in $dq \le l_{secret}(\lambda + O(\log n + \log l_{secret}))$. Thus the scheme is poly-time in $n^2 l_{secret}(\lambda + O(\log n + \log l_{secret}))$, or simplifying, in $n\lambda l_{secret}$.

Let's calculate the share size:

$$\begin{aligned} l_{share} &\le \log |(\text{set of shares})| \\ &= \log |K \times F^{3n}| \\ &= l_{secret} + 3nq \\ &= l_{secret} + 3n(\lambda + O(\log n + \log l_{secret})) \\ &= l_{secret} + 3n\lambda + n \cdot O(\log n + \log l_{secret}). \end{aligned}$$

□

Proposition 4.11. *The SS scheme $\Omega^{RB}(n, K, \lambda) = (n, K, K \times F^{3n}, SH, RE)$ is t-threshold and λ-robust.*

Proof. The property of t-threshold is inherited directly from Shamir scheme. Now, since the adversary knows nothing about the randomly generated keys of the honest players, proposition B.4 states that for any adversary strategy the probability that a given honest player accepts the modified share of a given corrupted player is at most $\frac{d}{2^q}$. By union bound, the probability that any of the $t + 1$ honest players accepts any of the at most t modified shares is $\le t(t+1)\frac{d}{2^q} < \frac{n^2 d}{2^{q+2}}$. If no honest player is deceived, all of them will recover the correct secret, and this secret will be chosen by majority vote and succesfully outputted. All that is left is to prove is that $\frac{n^2 d}{2^{q+2}} \le 2^{-\lambda}$.

If we omit the trivial case $n = 1$ (where the failure probability is zero), then we can also rule out some trivial low values of $|K|$ and q, and obtain the inequality $d = \left\lceil \frac{\log |K|}{q} \right\rceil \le \log |K|$. On the other hand, $q \ge \lambda + 2\log n + \log\log |K| - 2$, so $\frac{n^2 d}{2^{q+2}} \le \frac{n^2 \log |K|}{2^\lambda n^2 \log |K|} = 2^{-\lambda}$. □

4.4 CDF Scheme

We present the RSS scheme proposed by Cramer, Damgård and Fehr in 2001 [5], based on ideas from Cabello, Padró and Sáenz [4].

Recall that in the Rabin Ben-Or scheme, the dealer first computes the shares of the secret, and then uses a MAC to produce keys and tags for each pair of players. Thus in the reconstruction phase, any player can judge what the good shares are, and then reconstruct with them a secret.

In the CDF scheme, the dealer produces just one key and one tag from the secret, and then computes shares for these three values. Now, in the reconstruction phase any subset of $t+1$ players can reconstruct a secret, and then judge if it is good.

The (key, tag) couple we use and the linearity of Shamir scheme allow us to isolate completely the effect of the adversary strategy in the reconstruction of the secret.

Scheme 4.12 (CDF)**.** For a positive odd integer $n = 2t+1$, a real number $\lambda > 1$, and a finite field K satisfying $|K| \geq 2^{n+\lambda}$, we define the CDF SS scheme $\Omega^{CDF}(n, K, \lambda) = (n, K, K^3, SH, RE)$ as:

SH For input $s \in K$, pick a random key $\kappa \in K$, and compute the tag $\tau = s \cdot \kappa$. Next, using Shamir scheme $\Omega^S(t, n, K)$ (scheme 3.12), generate vectors $(s_i)_{i\in[n]}$, $(\kappa_i)_{i\in[n]}$ and $(\tau_i)_{i\in[n]}$ from each of the previous values, respectively. Output the shares $(s_i, \kappa_i, \tau_i)_{i\in[n]}$.

RE Let the input shares be $(s'_i, \kappa'_i, \tau'_i)_{i\in[n]}$. For every set A of $t+1$ shares different from $\perp$, do the following:

Reconstruction according to A: Use A and Shamir scheme to reconstruct the respective values s_A, κ_A and τ_A; and if they hold the equality $\tau_A = s_A \cdot \kappa_A$, then label s_A as *accepted*.

If there is a unique element $s' \in K$ labeled as accepted, output s', otherwise output $\perp$.

Proposition 4.13. *The SS scheme $\Omega^{CDF}(n, K, \lambda) = (n, K, K^3, SH, RE)$ is t-threshold and λ-robust.*

Proof. The property of t-threshold is inherited directly from Shamir scheme. Suppose that the original values of the secret, the key and the tag are s, κ and τ respectively. For any adversary strategy, RE will receive at least $t+1$ original shares (from the honest players), so the correct secret s will always be labeled as accepted. It remains to prove that the probability that another secret is accepted is at most $2^{-\lambda}$.

Fix a set A of $t+1$ shares different from $\perp$, let s_A, κ_A and τ_A be the corresponding values reconstructed through Shamir, and suppose that $s_A \neq s$. For this fixed set, the linearity of Shamir scheme implies that the adversary strategy determines precisely the values of the differences $\Delta s = s - s_A \neq 0$, $\Delta\kappa = \kappa - \kappa_A$ and $\Delta\tau = \tau - \tau_A$. Then:

$$\begin{aligned} P[s_A \text{ is accepted}] &= P[\tau_A = s_A\kappa_A \mid \tau = s\kappa] \\ &= P[\tau - \Delta\tau = (s-\Delta s)(\kappa - \Delta\kappa) \mid \tau = s\kappa] \\ &= P[-\Delta\tau = -s\Delta\kappa - \kappa\Delta s + \Delta s\Delta\kappa \mid \tau = s\kappa] \\ &= P\left[\kappa = \frac{\Delta\tau - s\Delta\kappa}{\Delta s} + \Delta\kappa\right]. \end{aligned}$$

But κ is chosen at random from K independently from anything else, so even if we assume that the adversary knows the secret s (this assumption could only improve his strategy), the probability that κ holds the equality above (and that s_A is accepted) is at most $|K|^{-1} = 2^{-n-\lambda}$. By union bound, the probability that $s_A \neq s$ is accepted, for any set A of $t+1$ shares, is at most $\binom{n}{t+1}2^{-n-\lambda} < 2^{-\lambda}$. This completes the proof. □

How convenient is the CDF scheme? Its main feature is its share size, which is evidently $l_{share} = 3l_{secret} \geq 3(n+\lambda)$. If the scheme is used for a very small set of secrets, where $l_{secret} \approx n + \lambda$, then we obtain the desired reduced share size $l_{share} = l_{secret} + O(n+\lambda)$. However if $l_{secret} \gg n + \lambda$, as is the case in most applications, the share size is rocketing high, while the failure probability is much smaller than required. The scheme has been corrected by Cramer, Dodis, Fehr, Padró and Wichs in [6]. In the improved version of the scheme, the correlation between the secret size and the security parameter has been eliminated, so the share size is $l_{share} = l_{secret} + O(n+\lambda)$ (omitting logarithms) for all values of the variables.

On the other hand, there is no known reconstruction algorithm that is poly-time on the number of players. For instance, the reconstruction algorithm above makes a Shamir reconstruction for potentially every set of $t+1$ players, and the number of such sets can be proved to be $\binom{n}{t+1} \geq \frac{2^{n-1}}{\sqrt{n}}$. This function cannot be upper-bounded by a polynomial expression in n. Another drawback of the scheme is that it only works for a model where the adversary does not have access to the communication channels between the honest players and the reconstructor (see remark 4.1).

5 Reducing the Share Size in RSS

We refer to our new RSS scheme simply as the *New* scheme, for want of a better label.

5.1 New Scheme

This RSS scheme follows the same design as the Rabin Ben-Or scheme (4.9), except that the reconstruction algorithm is improved, leading the scheme to reduce its failure probability, and consequently the finite field used for the MAC can be chosen of smaller size. As in the Rabin Ben-Or scheme, in the sharing phase the dealer generates keys and tags to allow every player to verify the authenticity of the share of every other player.

In the reconstruction phase, the reconstructor R no longer focuses on the shares accepted by a particular player, but on the number of the players that accept a particular share, in a voting system. Through several rounds, R eliminates the players whose shares receive low votes, and eliminates also the votes given by these players, to end up with a core of *certified* players, who support each other with their votes. Finally, R uses an error-correcting code (ECC, section A) to raise the chances of recovering the original secret.

The joint use of the elimination rounds and the ECC is very effective: an adversary attack that modifies a large number of shares will be greatly affected by the rounds, while an adversary attack with too few modified shares will be neutralized by the ECC.

Scheme 5.1 (New)**.** For a positive odd integer $n = 2t + 1$, a finite field K satisfying $|K| > n$, and a real number $\lambda > 1$, we define the New SS scheme $\Omega^N(n, K, \lambda) = (n, K, K \times F^{3n}, RE, SH)$, in the following way:

Define the real number $A = (e2^\lambda)^{2/(t+1)}$ and the integers $q = \lceil \log(t+1) + \log A + \log\log|K| \rceil$ and $d = \left\lceil \frac{\log|K|}{q} \right\rceil$. Let F be the field of order 2^q, and fix an injective function ψ from K to F^d. Fix also $n+1$ distinct points $x_0, x_1, \cdots, x_n$ in K, and consider the Shamir SS scheme $\Omega^S(t, n, K)$ that is defined for those points (scheme 3.12). Define the algorithms:

SH For input secret $s \in K$, use the Shamir scheme to generate the vector $(s_i)_{i\in[n]} \in K^n$. For all indices $i, j \in [n]$, pick a random key $\kappa_{ij} \in F^2$, and use the polynomial-evaluation MAC (definition B.3) to define the tag $\tau_{ji} = h_{F,d}(\psi(s_j), \kappa_{ij}) \in F$. Output $(s_i, \kappa_{i1}, \cdots, \kappa_{in}, \tau_{i1}, \cdots, \tau_{in})_{i\in[n]}$.

RE Let the input shares be $(s'_i, \kappa'_{i1}, \cdots, \kappa'_{in}, \tau'_{i1}, \cdots, \tau'_{in})_{i\in[n]}$. For all indices $i, j \in [n]$, check whether $\tau'_{ji} = h_{F,d}(\psi(s'_j), \kappa'_{ij})$, in which case we say that i *accepts* j. Let $I = [n]$.

Elimination round: Eliminate from I all indices j with the property that j is accepted by strictly fewer than $t + 1$ indices i in I. If some index is indeed eliminated, start a new elimination round (with the updated I).

ECC reconstruction with certified shares: Let $S = \{x_i \mid i \in I\}$, and use Reed-Solomon code $RS_{K,S}[|I|, t+1] = (C, D)$ (definition A.1) to compute $g(X) = D\left((s_i)_{i\in I}\right) \in K[X]_{\leq t}$ (which is the polynomial satisfying $g(x_i) = s'_i$ for a maximum number of indices i). Output $g(x_0)$.

Proposition 5.2. *The scheme $\Omega^N(n, K, \lambda)$ is poly-time in $n\lambda l_{secret}$, and its share size is $l_{share} \leq l_{secret} + 12\lambda + n \cdot O(\log n + \log l_{secret})$.*

Proof. The running time of the elimination rounds is bounded by $O(n^3)$, because there are at most $t+1 \leq n$ rounds, and in each round we read at most $|I|^2 \leq n^2$ Boolean values of the type "i accepts j"; so this part of the reconstruction algorithm is poly-time in $n\lambda l_{secret}$. The rest of the scheme is composed of parts already proven to be poly-time in $n\lambda l_{secret}$. Let's calculate the share size:

$$\begin{aligned} l_{share} &\leq \log |(\text{set of shares})| \\ &= \log |K \times F^{3n}| \\ &= l_{secret} + 3nq \\ &= l_{secret} + 3n[\log A + O(\log n + \log l_{secret})] \\ &\leq l_{secret} + 3n \left[\frac{4}{n}(\lambda + O(1)) + O(\log n + \log l_{secret})\right] \\ &= l_{secret} + 12\lambda + n \cdot O(\log n + \log l_{secret}). \end{aligned}$$

□

Proposition 5.3. *The SS scheme $\Omega^N(n, K, \lambda) = (n, K, K \times F^{3n}, RE, SH)$ is t-threshold and λ-robust.*

It is convenient at this point to introduce some notation. The subset of players who pass the elimination rounds will be called *certified*. We distinguish between the two components of a share: the *Shamir share* and the *redundancy part*, the latter composed of n keys and n tags. Likewise we identify two kinds of corrupted players: an *active* one provides a bogus Shamir share in the reconstruction phase, and a *passive* one provides an unmodified Shamir share. Both kinds of corrupted players may provide modified redundancy parts.

Lemma. *The reconstruction algorithm of this scheme fails only if, among the corrupted players that are certified, a strict majority is active.*

Proof of Lemma. Remember that we assume in our model that there are at least $t + 1$ honest players. Notice that all of them will be certified, for any adversary strategy, because they all accept each other's Shamir shares. Let t' be the number of corrupted players that are certified ($0 \leq t' \leq t$). In the final step of the algorithm, the Shamir shares of the certified players are regarded as the Reed-Solomon coding of a $(t+1)$-symbol message with t' added redundancy symbols. By proposition A.2, the message (and the secret) can be recovered whenever at most $\lfloor t'/2 \rfloor$ Shamir shares are bogus. The lemma follows. □

Proof of Proposition 5.3. The property of t-threshold is directly inherited from Shamir scheme, so we go straight to proving robustness. The adversary knows nothing about the randomly generated keys of the honest players, thus by proposition B.4 the probability that a given honest player accepts the Shamir share of a given active corrupted player is at most $\frac{d}{2^q}$. If we omit the trivial case $n = 1$ (where the failure probability is zero), then we can also rule out some trivial low values of $|K|$ and q, and obtain the inequality $d = \left\lceil \frac{\log |K|}{q} \right\rceil \leq \log |K|$. On the other hand, $2^q \geq (t + 1)A \log |K|$, so $\frac{d}{2^q} \leq \frac{1}{(t+1)A}$.

Since all honest players accept all good Shamir shares, then all passive players will be certified. In order to optimize the adversary strategy, we may assume that all corrupted players modify their redundancy parts to accept each other's Shamir shares, and that all t of them participate in the reconstruction phase (i.e. no share is $\perp$). For the latter assumption, we reason that any strategy where some corrupted players do not participate is weaker than a strategy where they are active: in case they are eliminated, the two strategies are equivalent. Finally, notice that whether or not the corrupted players modify their redundancy parts to reject the honest players' Shamir shares is irrelevant.

We define the following closely related terms: P^F is the maximum of the failure probabilities $P^F(s, I, MO)$ over all secrets $s \in K$, and all adversary strategies (I, MO); P_a^F is the same maximum, but only over strategies with a active and $t-a$ passive players; and $P_{a,c}^F$ is the same as P_a^F, but with the added requirement that exactly c out of the a active players become certified. By the previous lemma, $P_{a,c}^F$ is non-zero only for the values $t/2 < a \leq t$ and $t - a < c \leq a$, and we obtain the following formulas:

$$P_a^F = \sum_{t-a<c\leq a} P_{a,c}^F, \tag{1}$$

$$P^F = \max_{t/2<a\leq t} P_a^F. \tag{2}$$

Now, let's find a bound for $P_{a,c}^F$, $t-1 < c \leq a$. Notice that if exactly c out of a active players are certified, then each one of their bogus Shamir shares must be accepted by at least $a-c+1$ honest players. Indeed, each one of them must be accepted by at least $t+1$ certified players, of which at most $(t-a)+c$ are corrupted ($(t-a)$ passive and c active). Hence:

$$\begin{aligned}
P_{a,c}^F &\leq P[\text{Exactly } c \text{ out of } a \text{ active players are certified}] \\
&\leq P[\text{Exactly } c \text{ bogus shares are accepted by } \geq a-c+1 \text{ honest players}] \\
&\leq \binom{a}{c} P[c \text{ specific bogus shares are accepted by } \geq a-c+1 \text{ honest pl.}] \\
&= \binom{a}{c} \left(P[\text{a specific bogus share is accepted by } \geq a-c+1 \text{ honest pl.}]\right)^c \\
&\leq \binom{a}{c} \left[\binom{t+1}{a-c+1}\left(\frac{1}{(t+1)A}\right)^{a-c+1}\right]^c \\
&\leq \binom{a}{c} \left[\frac{(t+1)^{a-c+1}}{(a-c+1)!}\left(\frac{1}{(t+1)A}\right)^{a-c+1}\right]^c \\
&= \binom{a}{c} \frac{A^{-c(a-c+1)}}{((a-c+1)!)^c} \\
&= \frac{A^{-c(a-c+1)}}{((a-c)!)^c}\left[\frac{a!/(a-c)!}{c!(a-c+1)^c}\right] \\
&= \frac{A^{-c(a-c+1)}}{((a-c)!)^c} \prod_{1\leq i\leq c} \underbrace{\frac{a-c+i}{i(a-c+1)}}_{\leq 1}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{A^{-c(a-c+1)}}{((a-c)!)^c} \\
&\leq \frac{A^{-c(a-c+1)}}{(a-c)!} \\
&\leq \frac{1}{(a-c)!} \max_{1\leq i\leq a} A^{-i(a-i+1)} \\
&= \frac{A^{-a}}{(a-c)!},
\end{aligned}
$$

where in the last line we used the fact that $A = \left(e2^{\lambda}\right)^{\frac{2}{t+1}}$ is constant and greater than 1, and that for $1 \leq i \leq a$ the exponent $-i(a-i+1)$ attains its maximum at either $i = 1$ or $i = a$.

Now, using equation (1), we compute a bound for P_a^F:

$$
\begin{aligned}
P_a^F &= \sum_{t-a<c\leq a} P_{a,c}^F \\
&\leq \sum_{t-a<c\leq a} \frac{A^{-a}}{(a-c)!} \\
&\leq A^{-a} \sum_{i\geq 0} \frac{1}{i!} = eA^{-a}.
\end{aligned}
$$

And finally, equation (2) gives:

$$
\begin{aligned}
P^F &\leq \max_{t/2<a\leq t} P_a^F \\
&\leq \max_{t/2<a\leq t} eA^{-a} \\
&\leq eA^{-\left(\frac{t+1}{2}\right)} \\
&= \leq e\left[\left(e2^{\lambda}\right)^{\frac{2}{t+1}}\right]^{-\left(\frac{t+1}{2}\right)} = 2^{-\lambda}.
\end{aligned}
$$

□

We state formally what we just proved.

Theorem 5.4. *For a positive odd integer $n = 2t+1$, a finite field K satisfying $|K| > n$, and a real number $\lambda > 1$, there is a secret sharing scheme $\Omega^N(n, K, \lambda)$ for n players and a secret size $l_{secret} = \log|K|$, that is t-threshold, λ-robust and poly-time in $n\lambda l_{secret}$, and with share size:*

$$
l_{share} \leq l_{secret} + 12\lambda + n \cdot O(\log n + \log l_{secret}).
$$

6 Conclusions and Open Questions

We proved that for a threshold SS scheme, the share size is lower-bounded by the secret size. Then, we examined the existence of robust secret sharing schemes depending on the number of corrupted players, the conclusion being that the interesting case is the range between one third and one half, because outside that interval robust secret sharing is either impossible or already ideal. We focused thus on the scenario where an m-bit secret is shared among $n = 2t + 1$ players, of which t are corrupted (the maximum possible number), and the failure probability is required to be at most $2^{-\lambda}$.

For this scenario, we presented two common RSS schemes: one by Rabin and Ben-Or ([13]), and one by Cramer, Damgård and Fehr ([5]), and then we introduced a new RSS scheme. We compared their performances, and the results, with omitted logarithmic terms, are as follows: The Rabin Ben-Or scheme is poly-time in $m\lambda n$ (i.e. it is efficient), but its share size is $m + O(\lambda n)$. The CDF scheme has a smaller share size, $m + O(\lambda + n)$, but it has no known efficient reconstruction algorithm. Finally, the new scheme is poly-time in $m\lambda n$ and has a reduced share size of $m + O(\lambda + n)$.

The new scheme is based on Rabin Ben-Or scheme, i.e. it splits the secret using Shamir SS scheme, and adds to the shares a redundancy part, allowing each player to verify the authenticity of each other player's share. The redundancy parts of the shares are constituted by keys and tags, and come from the application of a polynomial-evaluation message authentication code to every pair of players. The improvement in the new scheme lies in the reconstruction algorithm. It features a series of elimination rounds that detect a great deal of bogus shares, followed by a special reconstruction that is based on Reed Solomon error correcting codes.

In the new scheme, the proof of its robustness considers all possible scenarios, depending on the initial number of bogus Shamir shares, and the number of them after the elimination rounds. We proved that for any such scenario the reconstruction algorithm may fail only when there are at least $n/4$ deceptions, i.e. $n/4$ pairs $\langle P_i, s'_j \rangle$ of an honest player P_i accepting a bogus share s'_j. By contrast, in the Rabin Ben-Or scheme the proof of its robustness shows that the reconstruction algorithm may fail only when there is at least one deception. Then, in both cases the security parameter $2^{-\lambda}$ turns out to be the probability that $n/4$ or 1 deceptions occur, respectively.

If we compare the share sizes of both schemes, we notice that the gain factor $n/4$ is reflected transparently in them: $m + 3n\lambda + n \cdot O(\log n + \log m)$ is reduced to $m + 12\lambda + n \cdot O(\log n + \log m)$.

It is, of course, natural to wonder what a tight lower bound for the share size is, among the RSS schemes for the scenario above specified, but it seems clear that one cannot hope for anything better than $m + O(\lambda)$. Thus, it would be interesting to know how much the factor 12 can be reduced, and whether or not there is a scheme with a share size independent from n.

A Error-Correcting Codes

Error-correcting codes are special ways to represent data, so that the original information can be recovered even if some parts of it are corrupted. The basic idea is to *code* the original message through a specific algorithm that introduces some redundancy into it. If the corruption of the coded message is small enough, the added redundancy will allow the original information to be recovered, through a *decoding* process. Error-correcting codes are heavily applied in communications and data storage, and their utility in RSS is evident as well. The pioneer in this field of study was Richard Hamming [8].

A.1 Reed-Solomon Codes

We will describe the error-correcting codes introduced in 1960 by Irvin Reed and Gus Solomon [14]. These codes have very convenient properties, for instance they are *linear* and *maximum distance separable* (MDS). A precise definition of these and other related terms can be read in any coding theory book, such as the one by Van Lint [10], but in simple words the MDS property gives a notion of optimality in the tradeoff between the amount of redundancy added and the number of errors that can be corrected. The Reed-Solomon codes are closely related to Shamir SS scheme (3.12). As such, these two concepts can be naturally paired and used as the base of RSS schemes.

Definition A.1 (Reed-Solomon Code)**.** For integers, k, n, and a finite field K, satisfying $0 < k \leq n < |K|$, and a set of n distinct elements $S = \{x_1, \cdots, x_n\} \subset K$, we define the *Reed-Solomon code* $RS_{K,S}[n,k] = (C, D)$ as the pair of the *coding function* $C : K[X]_{k-1} \to K^n$, and the *decoding function* $D : K^n \to K[X]_{k-1}$, defined as follows:

C: For message $f(X) \in K[X]_{k-1}$, let $C\left(f(X)\right) = \left(f(x_i)\right)_{i\in[n]}$.

D: For vector $(y_i)_{i\in[n]}$ in K^n, find a polynomial $g(X) \in K[X]_{k-1}$ such that $g(x_i) = y_i$ for a maximum number of indices i, and let $D\left((y_i)_{i\in[n]}\right) = g(X)$.

Proposition A.2. *Consider the Reed-Solomon code $RS_{K,S}[n,k] = (C, D)$. For any message $f(X) \in K[X]_{k-1}$ and any vector $V \in K^n$, if V differs from $C\left(f(X)\right)$ in at most $\left\lfloor \frac{n-k}{2} \right\rfloor$ coordinates, then $D(V) = f(X)$.*

Proof. Let $f(X) \in K[X]_{k-1}$ and $V \in K^n$ be as above, i.e. V differs from $C\left(f(X)\right)$ in at most $\left\lfloor \frac{n-k}{2} \right\rfloor$ coordinates. To prove that $D(V) = f(X)$, it's enough to show that no other polynomial $g(X) \in K[X]_{k-1}$ has an image $C\left(g(X)\right)$ with so many coordinates in common with V.

Suppose then that for some $g(X) \in K[X]_{k-1}$, V also differs from $C\left(g(X)\right)$ in at most $\left\lfloor \frac{n-k}{2} \right\rfloor$ coordinates. As a consequence, $C\left(g(X)\right)$ differs from $C\left(f(X)\right)$ in at most $n-k$ coordinates, or equivalently, $g(x_i) = f(x_i)$ for at least k indices i. By Theorem 3.11, it must be that $g(X) = f(X)$. □

We can identify the message set $K[X]_{k-1}$ with K^k (through Theorem 3.11), and think of the coding function as adding $n-k$ redundancy symbols to a k-symbol message. Thus, the previous result says that the number of symbols that can be corrected corresponds to one half of the amount of redundancy.

In the definition of the decoding function above, it is not specified how to find the desired polynomial, and in fact the design of an efficient decoding algorithm is not at all trivial. In 1960, Peterson invented the first efficient algorithm for decoding Reed Solomon codes [12], and currently the Berlekamp-Welch algorithm, covered under US Patent [19], is one of the most widely used. A simplified version of the Berlekamp-Welch algorithm, provided by Gemmell and Sudan, can be found in [7].

Proposition A.3. *The Reed-Solomon code $RS_{K,S}[n,k] = (C,D)$ admits a decoding algorithm that is poly-time in $n \log |K|$.*

B Message Authentication Codes

A message authentication code (MAC) is an algorithm that generates a block of data (called a tag) based on a message and a determined secret key, so that it is attached to the message in order to improve the security of a communication. In particular, the two security properties involved in message authentication are *source authentication*, which prevents the acceptance of messages from a fraudulent source, and *data integrity*, which protects the data from modification. Confidentiality of the message is not required. As in the rest of this paper, in this section we focus on information-theoretic security.

The model works as follows. Player A wants to send a message m to player B over an insecure channel, and player C may modify it. However, A and B are assumed to share a secret (and usually randomly generated) key κ unknown to C, and they agree upon a MAC based on κ, to add security to the communication. A computes the tag $\tau = MAC(m, \kappa)$ and sends (m, τ) to B. Upon arrival of the (possibly modified) pair (m', τ'), B checks if $\tau' = MAC(m', \kappa)$, and accepts it accordingly. The MAC should permit B to correctly accept the message whenever it was unmodified, and reject a modified message with a very high probability.

MAC's with good cryptographic properties were introduced in the 1970s, and the first reference to a MAC algorithm is a 1972 US patent application by Simmons *et al.* (reference 10. in [15]). Several different terms are used in literature to define the specific cryptographic properties that a MAC should have (see for instance [11] and [17]), like being *collision free* and *pre-image resistant*. We specify below an *ad hoc* property that matches our needs in this paper.

Definition B.1. Given a *message set* $\mathcal{M}$, a *key set* $\mathcal{K}$ and a *tag set* $\mathcal{T}$, a corresponding *message authentication code* (MAC) is a function $h : \mathcal{M} \times \mathcal{K} \to \mathcal{T}$.

Definition B.2. Consider a MAC $h : \mathcal{M} \times \mathcal{K} \to \mathcal{T}$ and a positive real number λ. The MAC is λ-*secure* if for any two distinct messages $m \neq m' \in \mathcal{M}$, any two tags $\tau, \tau' \in \mathcal{T}$, and a random variable $\kappa \in \mathcal{K}$, the conditional probability $P\left[h(m', \kappa) = \tau' \mid h(m, \kappa) = \tau\right] \leq 2^{-\lambda}$.

In other words, if player C has zero information about the secret key, then for any strategy and any computational capabilities of C, the probability that B accepts a modified message is at most $2^{-\lambda}$.

B.1 Polynomial-Evaluation MAC

Many popular MAC's are based on *polynomial evaluation*. A thorough study of them is layed out by Bernstein in [1]. We present an optimized polynomial-evaluation scheme that uses a very small key, and can achieve highly efficient implementations based on Horner's rule. It was introduced in the early 1990's independently by den Boer [3], by Johansson, Kabatianskii, and Smeets [9], and by Taylor [18].

Definition B.3 (Polynomial-Evaluation MAC)**.** For a finite field F and an integer $0 < d < |F|$, the *polynomial-evaluation MAC* $h_{F,d} : F^d \times F^2 \to F$ is defined in the following way:

For a message $m = (m_1, \cdots, m_d) \in F^d$ and a key $\kappa = (a, b) \in F^2$ the corresponding tag is $\tau = h_{F,d}(m, \kappa) = a + \sum_{i=1}^{d} m_i b^i$.

Proposition B.4. *The polynomial-evaluation MAC $h_{F,d}$ is $(\log|F| - \log d)$-secure.*

Proof. Fix two distinct messages $m = (m_1, \cdots, m_d)$ and $m' = (m'_1, \cdots, m'_d)$ in F^d and two tags $\tau, \tau' \in F$, and let $\kappa = (a, b) \in F^2$ be a key generated at random. Define the differences $\Delta\tau = \tau' - \tau$ and $\Delta m_i = m'_i - m_i$, $\forall i \in [d]$. Then:

$$\begin{aligned}
&P\left[\tau' = h_{F,d}(m', \kappa) \mid \tau = h_{F,d}(m, \kappa)\right] \\
&= P\left[\tau' = a + \sum_{i=1}^{d} m'_i b^i \mid \tau = a + \sum_{i=1}^{d} m_i b^i\right] \\
&= P\left[\Delta\tau = \sum_{i=1}^{d} \Delta m_i b^i \mid \tau = a + \sum_{i=1}^{d} m_i b^i\right] \\
&= P\left[f(b) = 0 \mid \tau = a + \sum_{i=1}^{d} m_i b^i\right] \\
&= P\left[f(b) = 0\right]
\end{aligned}$$

Where we define $f(X) \in F[X]_{\leq d}$ as $f(X) = -\Delta\tau + \sum_{i=1}^{d} \Delta m_i X^i$, and in the last line we use the fact that, conditioned on the event $\tau = a + \sum_{i=1}^{d} m_i b^i$, b is still uniformly distributed. As $f(X)$ cannot be the zero polynomial (because $m \neq m'$), it has at most d distinct roots, and the probability that b is one of them is at most $\frac{d}{|F|} = 2^{\log d - \log|F|}$. This completes the proof. □

Proposition B.5. *The polynomial-evaluation MAC $h_{F,d}$ admits an algorithm that is poly-time in $d \log|F|$.*

References

[1] D.J. Bernstein. *Polynomial evaluation and message authentication*. 2007. URL: `http://cr.yp.to/papers.html#pema`.

[2] George Blakley. *Safeguarding Cryptographic Keys*. Proceedings of the National Computer Conference, 48, pp 313-317, 1979.

[3] B. den Boer. *A Simple and Key-Economical Unconditional Authentication Scheme* Journal of Computer Security, 2, pp 65-71, 1993.

[4] S. Cabello, C. Padró and G. Sáenz. *Secret Sharing Schemes with Detection of Cheaters for a General Access Structure*. 12th International Symposium on Fundamentals of Computation Theory, FCT'99, pp 185-193, 1999.

[5] R. Cramer, I. Damgård and S. Fehr. *On the Cost of Reconstructing a Secret, or VSS with Optimal Reconstruction Phase*. Advances in Cryptology - CRYPTO '01, pp 522-523, 2001.

[6] R. Cramer, Y. Dodis, S, Fehr, C. Padró and D. Wichs. *Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors*. Theory and Application of Cryptographic Techniques - EUROCRYPT '08, pp 471-488, 2008.

[7] P. Gemmell and M. Sudan. *Highly Resilient Correctors for Polynomials*. Information Processing Letters, 43(4):169–174, 1992.

[8] R.W. Hamming. *Error Detecting and Error Correcting Codes*. Bell System Technical Journal, 29(2):147-160, 1950.

[9] T. Johansson, G. Kabatianskii and, B.J.M. Smeets. *On the Relation Between A-Codes and Codes Correcting Independent Errors*. Advances in Cryptology - EUROCRYPT '93, pp 1-11, 1993.

[10] J.H. van Lint. *Introduction to Coding Theory*. Springer-Verlag (Graduate Texts in Mathematics series), 3rd ed., 1999.

[11] A. Menezes, P. van Oorschot, S. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997.

[12] W.W. Peterson. *Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes*. IRE Transactions on Information Theory, 6, pp 459–470, 1960.

[13] T. Rabin and M. Ben-Or. *Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (Extended Abstract)*. Proceedings of STOC '89, pp 73-85, 1989.

[14] I.S. Reed and G. Solomon. *Polynomial Codes over Certain Finite Fields*. SIAM Journal on Applied Mathematics, 8(2):300–304, 1960.

[15] G.J. Simmons. *How to Insure That Data Acquired to Verify Treaty Compliance are Trustworthy*. Contemporary Cryptology (IEEE), pp. 617-630, 1992.

[16] A. Shamir. *How to Share a Secret*. Communications of the ACM, 22(11): 612-613, 1979.

[17] C.R. Stinson. *Cryptography: Theory and Practice*. Chapman & Hall/CRC, 3rd Ed., 2006.

[18] R. Taylor. *An Integrity Check Value Algorithm for Stream Ciphers*. Advances in Cryptology - CRYPTO '93, pp 40-48, 1994.

[19] L. R. Welch and E. R. Berlekamp. *Error Correction of Algebraic Block Codes*. U.S. Patent Number 4,633,470, 1986.