Codes and Lattices in Cryptography Mini-Course **FCT-UNL, Spring 2023**

Notes 5: Information-theoretic cryptography

Lecturer: João Ribeiro

Introduction

These notes cover some more basic aspects of information-theoretic cryptography – protocols that are secure against computationally-unbounded adversaries.

5.1 Secret sharing

Consider the following scenario: We wish to distribute sensitive information s among n parties in a way that only select *authorized* subsets are able to reconstruct s, while *unauthorized* subsets of parties gain no information about the secret. For example, in order to ensure reliability and privacy, it could be useful to store a secret encryption key s across 10 servers in such a way that s can be reconstructed even if any 5 servers go offline, but also any powerful adversary that gains access into 4 or fewer servers learns no information about s. *Secret sharing*, a formalization of this concept, was introduced concurrently by Blakley [Bla79] and Shamir [Sha79].

A secret sharing scheme is composed of a probabilistic sharing procedure Share and a deterministic reconstruction procedure Rec. On input a secret s, we obtain the n shares of s

$$S = (S_1, \ldots, S_n) \leftarrow \mathsf{Share}(s).$$

The reconstruction procedure receives as input a subset of shares, and attempts to reconstruct the secret. For a set $T \subseteq [n] = \{1, 2, \ldots, n\}$, we define the projection $S_T = (S_i)_{i \in T}$. More formally, the following should hold.

Definition 5.1 (Threshold secret sharing scheme) *A pair* $(\mathsf{Share}, \mathsf{Rec})$ *is a* t-out-of-n secret sharing scheme *if the following two properties hold:*

- ***Correctness:*** *For any set of parties* $T \subseteq [n]$ *of size* $|T| \geq t$ *and any secret* s *we have that*
$$\Pr[\mathsf{Rec}(\mathsf{Share}(s)_T) = s] = 1.$$
- ***Privacy:*** *For any set of parties* $T \subseteq [n]$ *of size* $|T| < t$ *and any two secrets* s *and* s' *we have that* $\mathsf{Share}(s)_T$ *and* $\mathsf{Share}(s')_T$ *are identically distributed.*

Besides being interesting objects on their own, secret sharing schemes have also found important applications in other cryptographic tasks. For example, secret sharing is a fundamental building block of secure multiparty computation protocols. For more on this, see the excellent book of Cramer, Damgård, and Nielsen [CDN15].

5.1.1 Shamir's secret sharing scheme

Arguably the most well known and widely used secret sharing scheme is due to Shamir [Sha79] and is based on polynomial interpolation. Suppose that we wish to share a secret among n parties. Fix a prime power $q > n$ and n non-zero points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q \setminus \{0\}$, where $\mathbb{F}_q$ denotes the finite field of order q. Fix also a threshold t, specifying the number of parties required to reconstruct the secret. On input a secret $s \in \mathbb{F}_q$, the sharing procedure $\mathsf{Share}(s)$ in Shamir's secret sharing scheme works as follows:

1. Sample coefficients $c_1, c_2, \ldots, c_{t-1} \leftarrow \mathbb{F}_q$ and consider the polynomial
$$f(x) = s + \sum_{i=1}^{t-1} c_i x^i$$
over $\mathbb{F}_q$ of degree at most $t-1$;
2. Compute the i-th share S_i as the evaluation $S_i = f(\alpha_i)$ for $i \in [n]$.

To reconstruct the secret given t shares $S_{i_j} = f(\alpha_{i_j})$, perform Lagrange interpolation on these evaluations to recover f, and then compute $f(0)$ to recover the secret.

Theorem 5.1 *Shamir's scheme as described above is a t-out-of-n secret sharing scheme.*

Proof: Correctness follows from the fact that we can reconstruct the polynomial f, which has degree at most $t-1$, from any t evaluations via Lagrange interpolation. Then, we may recover $s = f(0)$. Regarding privacy, fix any $t-1$ shares $S_{i_1}, \ldots, S_{i_{t-1}} \in \mathbb{F}_q$. It suffices to note that for each secret $s \in \mathbb{F}_q$ there is exactly one polynomial f_s of degree at most $t-1$ such that $f_s(0) = s$ and $f_s(\alpha_{i_j}) = S_{i_j}$ for all $j = 1, \ldots, t-1$. Therefore, every secret is equally likely given $S_{i_1}, \ldots, S_{i_{t-1}}$. ■

As we shall see next, Shamir secret sharing fits into a more general framework connecting certain linear codes to secret sharing schemes.

5.1.2 Secret sharing from MDS codes

In the previous notes we used Reed-Solomon codes to construct locally dense lattices with desirable properties. One useful property of Reed-Solomon codes is that they have optimal minimum distance for a given length and dimension – they are *maximum distance separable (MDS)*. We recall the general definition here.

Definition 5.2 (MDS code) *An $[n, r, d]_q$-code, where n is the length, r is the dimension, and d is the minimum distance, is said to be* maximum distance separable (MDS) *if $d = n - r + 1$.*

MDS codes satisfy the following useful property.

Lemma 5.1 *Suppose that $\mathcal{C}$ is an MDS $[n, r, d]_q$-code. Consider an arbitrary subset of coordinates $S \subseteq [n]$ of size r. Then,*

$$\mathcal{C}_S = \{c_S : c \in \mathcal{C}\} = \mathbb{F}_q^r.$$

In other words, for every vector $v \in \mathbb{F}_q^r$ and any set $S \subseteq [n]$ of size r there exists exactly one codeword $c \in \mathcal{C}$ such that $c_S = v$.

Proof: Fix an arbitrary ordered subset of coordinates $S = \{i_1, \ldots, i_r\} \subseteq [n]$. For each vector $(\beta_1, \ldots, \beta_r) \in \mathbb{F}_q^r$ we show that there exists exactly one codeword $c \in \mathcal{C}$ such that $c_{i_j} = \beta_j$ for all $j \in [r]$. Suppose not. Then, there exist distinct codewords $c, c' \in \mathcal{C}$ such that $(c - c')_{i_j} = 0$ for all $j \in [r]$. This implies that c and c' disagree on at most $n - r < d$ coordinates, contradicting the minimum distance of $\mathcal{C}$. ■

We show how to build a secret sharing scheme from any MDS code, generalizing Shamir's secret sharing scheme. Suppose that we wish to share a secret $s \in \mathbb{F}_q$ among n parties with threshold t. Let $\mathcal{C}$ be an arbitrary MDS $[n+1, t, d]_q$-code, meaning that $d = (n+1) - t + 1$. We proceed as follows:

1. Sample a codeword $c = (c_0, \ldots, c_n)$ uniformly at random from the subset

$$\mathcal{C}_s = \{c \in \mathcal{C} : c_0 = s\};$$

2. Set the i-th share as $S_i = c_i$ for $i \in [n]$.

To reconstruct the secret given shares S_{i_j} for $j \in [t]$, we find the unique codeword $c \in \mathcal{C}$ such that $c_{i_j} = S_{i_j}$ for all j. Then, we output c_0.

This approach to constructing secret sharing schemes is due to Massey [Mas95]. The proof of the following result follows the same lines as the proof of Theorem 5.1.

Theorem 5.2 *The scheme defined above is a t-out-of-n secret sharing scheme.*

Proof: Regarding correctness, it suffices to argue that revealing t symbols $S_{i_1}, \ldots, S_{i_t}$ of a codeword determine the whole codeword. Indeed, if two distinct codewords $c, c' \in \mathcal{C}$ satisfy $c_{i_j} = c'_{i_j} = S_{i_j}$ for all $j \in [t]$, then c and c' differ in at most $n + 1 - t < d = n + 1 - t + 1$ coordinates, contradicting the minimum distance of $\mathcal{C}$.

To see privacy, fix any $t - 1$ shares $S_{i_1}, \ldots, S_{i_{t-1}}$. Since $\mathcal{C}$ has dimension t and is MDS, Lemma 5.1 guarantees that for every secret s there exists exactly one codeword $c \in \mathcal{C}$ such that $c_0 = s$ and $c_{i_j} = S_{i_j}$. This means that every secret is equally likely given $S_{i_1}, \ldots, S_{i_{t-1}}$. ■

Remark 5.1 We can obtain Shamir's secret sharing scheme as a special case of this MDS-based approach by choosing our MDS code $\mathcal{C}$ to be a Reed-Solomon code, which we defined and also used in the previous notes to construct locally dense lattices.

5.1.3 Important properties of these schemes

Besides satisfying the basic definition of secret sharing, the secret sharing schemes based on MDS codes above satisfy additional useful properties.

Linearity. Using the schemes above, we can perform linear operations on secrets by having each party locally perform operations on their shares. For example, suppose that $(S_1, \ldots, S_n)$ are shares of s and $(S'_1, \ldots, S'_n)$ are shares of s' for the same threshold t. Then, each party can locally compute $S''_i = S_i + S'_i$ over $\mathbb{F}_q$, and the resulting shares $(S''_1, \ldots, S''_n)$ are a t-out-of-n sharing of $s'' = s + s'$.

Robustness. The schemes above allow correct reconstruction of the secret even when some parties act dishonestly and report incorrect shares, provided that enough shares are revealed and not too many shares are reported incorrectly

To see why, let $\mathcal{C}$ be an $[n+1, t, d]_q$-code and suppose that a codeword $c \in \mathcal{C}$ is corrupted by erasing n_e coordinates (where "e" stands for "erasures") and arbitrarily modifying n_f other coordinates (where "f" stands for "flips"), yielding a corrupted string z. Then, it is not hard to check that there is at most one codeword of $\mathcal{C}$ that could have originated z in this manner whenever $n_e + 2n_f < d$. In other words, in such a case we can correct the errors introduced in z and recover c.

Consider now a setting where $\mathcal{C}$ is MDS and t' parties reveal their shares, but $t'' \leq t'$ of them are dishonest and can reveal incorrect shares. This corresponds to the setting above with $n_e = n+1-t'$ and $n_f = t''$. Therefore, we correctly reconstruct the secret whenever

$$n_e + 2n_f = n + 1 - t' + 2t'' < d = n + 1 - t + 1.$$

This robustness constraint can be rewritten as

$$t'' < \frac{t' - t + 1}{2}.$$

A note on efficiency of robust reconstruction. While for every linear code there exists an efficient procedure that corrects erasures in codewords (i.e., missing shares), it is not straightforward to devise an efficient procedure that corrects flips for any given MDS code $\mathcal{C}$, which we would need to *robustly* reconstruct our secret in the secret sharing scheme defined by $\mathcal{C}$. But, for example, we do know of such efficient error-correcting procedures when $\mathcal{C}$ is a Reed-Solomon code (see [GRS22, Chapter 17]).

References

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318, 1979.

[CDN15] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[GRS22] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. 2022. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book.

[Mas95] James L. Massey. Some applications of coding theory in cryptography. *Codes and Ciphers: Cryptography and Coding IV*, pages 33–47, 1995.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.