Recitation 11: Secret Sharing and Interactive Proofs

1 Secret Sharing

The storage of a piece of information, or secret, in one centralized location can leave the secret vulnerable to theft by an adversary. Secret sharing is a method to distribute a secret across multiple parties in order to make theft of the secret difficult. Suppose we have a secret key $k$ that we wish to distribute among $n$ parties.

Definition 1. A $(t, n)$-threshold secret sharing scheme is a method to distribute the secret $k$ among $n$ parties such that:

- Each of the $n$ parties receives a share of the secret.
- Any subset of the $n$ parties with at least $t + 1$ members can recover the secret together.
- Any subset of the $n$ parties of size $\leq t$ have no information about the secret.

1.1 Sum Sharing

Suppose that we have a secret $k \in \mathbb{Z}$ such that $0 \leq k < M$, and suppose that $M$ is publicly known. We will propose a $(n - 1, n)$-threshold secret sharing scheme.

Scheme 1. SUM-sharing($k$):

- Choose a prime number $p$ such that $p > M$.
- Choose $x_1, \ldots, x_{n-1}$ each uniformly at random from $\mathbb{Z}_p$, and assign each $x_i$ to party $i$.
- Set $x_n = k - (x_1 + \ldots + x_{n-1}) \pmod{p}$, and assign $x_n$ to party $n$. 

1.2 Shamir Secret Sharing

Let $k$ be a random secret chosen from $\mathbb{Z}_p$. We wish to devise a $(t, n)$-threshold secret sharing scheme. Specifically, given $t$ or fewer shares of the secret, the probability of the secret being any element of $\mathbb{Z}_p$ must equal $1/p$, while $t + 1$ shares uniquely determine the secret.

**Scheme 2.** Shamir’s Secret Sharing($k, p, t, n$):

- Choose $t$ coefficients $R_1, \ldots, R_t$ uniformly at random from $\mathbb{Z}_p$.
- Let $P(x) = R_t x^t + R_{t-1} x^{t-1} + \ldots + R_1 x^1 + k \pmod{p}$.
- Assign $P(i)$ as the secret share to party $i$ for $1 \leq i \leq n$.

**Theorem 1.** Shamir’s secret sharing is a $(t, n)$-threshold secret sharing scheme.

**Proof.** First, we will show that knowledge of any $t + 1$ shares allows computation of $k = P(0)$. Recall that a polynomial with coefficients in $\mathbb{R}$ can be uniquely determined through evaluation at $t + 1$ distinct points. This result required only that coefficients of the polynomial have inverses, a condition that holds for any field, including $\mathbb{Z}$. Thus, knowledge of $t + 1$ shares allows direct computation of $P(x)$, a computation that can be achieved using Lagrange interpolation.

Next, we show that $t$ shares give no information about $P(0)$. Denote share $i$ by $y_i$. For any $k \in \mathbb{Z}_p$, there exists a unique polynomial $P(x)$ such that $P(0) = k$ and $P(i) = y_i$ for all $t$ available shares. Thus, for all sets of shares, for any potential secrets $k_1, k_2$, $\Pr[k = k_1] = \Pr[k = k_2]$. \qed

2 Interactive Proofs

2.1 IP Concepts

An interactive proof is a system consisting of two parties, a prover (Alice) and a verifier (Bob), who exchange messages regarding a theorem about an object $x$. Alice wishes to convince Bob that the theorem is true without necessarily giving him the key piece of information, called a *witness*, that directly proves the theorem. This is achieved probabilistically, so that for any
$\epsilon$, after sufficiently many rounds, Bob believes that there is no more than $\epsilon$
chance that Alice is lying to him.

The exchange consists of Alice sending Bob messages, Bob (potentially)
tossing coins to ask questions in response to those messages, Alice replying
with answers, and Bob finally deciding whether to accept or reject the proof.
We’ll now present a more formal definition - let’s say Alice and Bob choose
strategies $P$ and $V$ respectively. Then,

**Definition 2.** $(P, V)$ is an interactive proof of a predicate $L$ if completeness
and soundness are satisfied.

- Completeness: If $L(x) = True$, $\Pr[\text{output}_V(P, V_x) = \text{accept}] = 1$.
- Soundness: If $L(x) = False$, then for any $P' \neq P$, $\Pr[\text{output}_V(P', V_x) = \text{accept}] < 1/2$.

Conceptually, Bob must accept a true input, but reject a false input with
sufficiently high probability.

## 2.2 Quadratic Residuosity

We now present an interactive proof protocol for a particular problem, that
of determining quadratic residuosity.

Let $N$ be a prime number (in general, $N$ doesn’t have to be prime, but
it makes our analysis a bit simpler here, so we’ll assume it for these notes).

**Definition 3.** A number $x$ is a quadratic residue $\pmod{N}$ if there exists $y$
such that $y^2 \equiv x \pmod{N}$. We write this as $x \in QR_N$.

**Example 1.** Consider the case of $N = 13$. Then,

<table>
<thead>
<tr>
<th>$x$</th>
<th>$x^2 \pmod{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1$</td>
<td>$1$</td>
</tr>
<tr>
<td>$2$</td>
<td>$4$</td>
</tr>
<tr>
<td>$3$</td>
<td>$9$</td>
</tr>
<tr>
<td>$4$</td>
<td>$3$</td>
</tr>
<tr>
<td>$5$</td>
<td>$12$</td>
</tr>
<tr>
<td>$6$</td>
<td>$10$</td>
</tr>
</tbody>
</table>
Now, let \( x \in QR_N \). Prover \( P \) knows \( x, N \), a witness \( y \) such that \( y^2 \equiv x \) (mod \( N \)). The verifier \( V \) only knows \( x \) and \( N \), and does not know whether \( x \in QR_N \). The following protocol is an interactive proof for proving quadratic residuosity.

**Protocol 1. Quadratic Residuosity Protocol**

1. \( P \) chooses \( r \in \mathbb{Z}_n^\ast \) at random, computes \( s \equiv r^2 \) (mod \( N \)), and sends \( s \) to \( V \).
2. \( V \) tosses a coin to get \( b \in \{0, 1\} \), and sends \( b \) to \( P \).
3. \( P \) now computes \( z \). If \( b = 0 \), then \( z = r \). If \( b = 1 \), then \( z \equiv ry \) (mod \( N \)). \( P \) sends \( z \) to \( V \).
4. If \( b = 0 \), then \( V \) accepts if \( z^2 \equiv s \) (mod \( N \)). If \( b = 1 \), \( V \) accepts if \( z^2 \equiv sx \) (mod \( N \)).

**Theorem 2.** Protocol 1 is a valid IP for quadratic residuosity, i.e. it satisfies completeness and soundness.

*Proof.* First, we will show completeness, i.e. if \( x \in QR_N \) and the prover does not cheat, we will show that \( V \) will always accept. Consider both cases of the coin toss. If \( b = 0 \), then \( z = r \), so \( V \) will verify that \( z^2 = r^2 \equiv s \) (mod \( N \)). If \( b = 1 \), then \( V \) can verify that \( z^2 \equiv r^2y^2 \equiv sx \) (mod \( N \)).

Second, we will show soundness, i.e. if \( x \not\in QR_N \) and the prover cheats by choosing an alternate strategy \( P' \), the probability of \( V \) verifying is less than \( 1/2 \). In math,

\[
\Pr_{b \in \{0, 1\}} \left[ \text{output}_V(P', V, x, b) = \text{accept} \right] < \frac{1}{2}.
\]

This proof is a little bit more tricky. Since the prover can cheat, we aren’t guaranteed that she will send \( s \equiv r^2 \) in the first step. Thus, we consider two cases - \( s \) may or may not actually be a quadratic residue (mod \( N \)).

1. Assume \( s \in QR_N \). \( V \) will toss \( b = 1 \) with probability \( 1/2 \) - we will now just consider this case. Let \( z \) be the message sent by \( P' \) in Step 3. \( V \) will accept if and only if \( z^2 \equiv sx \) (mod \( N \)). However, if \( z^2 \equiv sx \), then

\[
x \equiv z^2s^{-1} \equiv z^2(r^2)^{-1} \equiv z^2(r^{-1})^2 \equiv (zr^{-1})^2 \quad \text{(mod \( N \))},
\]

implying that \( x \in QR_N \). This leads to a contradiction, so if \( b = 1 \), \( V \) is guaranteed to reject. Thus, \( \Pr[\text{accept}|s \in QR_N] < 1/2. \)
2. Assume $s \notin QR_n$. $V$ will toss $b = 0$ with probability $1/2$ - we will now just consider this case. If $b = 0$, $P'$ must find some $z$ such that $z^2 \equiv s \pmod{N}$, which is impossible by assumption. Thus, $\Pr[\text{accept}|s \in QR_N] < 1/2$.

\[\square\]

3 Error Correcting Codes (Optional)

3.1 General Information about ECCs

Suppose that we have a message $\sigma$ consisting of $n$ symbols that we wish to send to a receiving party. However, the only communication channel available to us is a noisy communication channel, and each symbol has some chance of being distorted in the process of transmission. Error-correcting codes provide us a way to introduce redundancy into the message, allowing the receiving party to decipher a partially altered message, as long as the communication channel isn’t too noisy.

Let $\Sigma$ be the alphabet from which symbols in our message are taken.

**Definition 4.** An *encoding* $E$ is function $E : \Sigma^n \to \Sigma^m$, where $m > n$. An encoding maps a message to a codeword.

**Definition 5.** A *decoding* $D$ is function $D : \Sigma^m \to \Sigma^n$, where $m > n$. A decoding maps a codeword to a message.

**Definition 6.** A *code word* is a member of the image of the encoding function. In our case, a code word has $m$ symbols.

Let $\Delta(\sigma, \tau)$ be the Hamming distance between two messages $\sigma, \tau$, i.e. the number of entries on which $\sigma$ and $\tau$ differ. We will now define the *distance* of a code, a measure of the robustness of a code to noise in the communication channel.

**Definition 7.** An encoding $E$ has *distance* $\alpha$ if for any distinct messages $\sigma, \tau \in \Sigma^n$, $\Delta(\sigma, \tau) \geq \alpha$.

**Definition 8.** A *t-error* is a transmission of a code word $E(\sigma)$ such that no more than $t$ symbols are altered across the channel.
Theorem 3. A code $E$ has distance $2t + 1$ if and only if it can correct all $t$-errors.

Proof. We begin with the if direction. Assume for the sake of contradiction that there exists a code $E$ with distance $2t$ that can correct all $t$-errors. Consider a distorted transmission that has Hamming distance of exactly $t$ from the two nearest valid code words, which is possible as a result of $E$ being a distance $2t$ code. Then, the code cannot unambiguously map the distorted transmission to either of the code words, so $E$ must have distance $2t + 1$.

Next, we show the only if direction. If a code $E$ has distance $2t + 1$, then any $t$-error transmission is within $t$ Hamming distance of exactly one valid code word. \qed

3.2 Reed-Solomon Codes

Reed-Solomon codes are a type of error correcting code that employ polynomials over finite fields to create a high distance code. We will choose our finite field $F$ in consideration be $\mathbb{Z}_p$ for some prime number $p$. Furthermore, for the purpose of simplicity in our Reed-Solomon description, we will assume that our alphabet $\Sigma = \mathbb{Z}_p$ as well, though this is not necessary for a general Reed-Solomon code.

Let $\sigma \in \Sigma^n$ be the message that we wish to transmit. We will represent $\sigma$ as a polynomial $f_\sigma : \mathbb{Z}_p \rightarrow \mathbb{Z}_p$. Let

\[ f_\sigma(x) = \sum_{i=0}^{n-1} \sigma_i x^i \pmod{p}. \]

Then, our encoding $E(\sigma)$ is defined as

\[ E(\sigma) = \{f_\sigma(x_1), f_\sigma(x_2), \ldots, f_\sigma(x_m)\} \]

where $x_1, \ldots, x_m$ are $m$ distinct values taken from $\mathbb{Z}_p$.

Theorem 4. A Reed-Solomon code $E : \Sigma^n \rightarrow \Sigma^m$ has distance $m - n + 1$.

Proof. Since $f_\sigma$ is a degree $n - 1$ polynomial, it can be interpolated from $n$ points. Thus, for any $m > n$, the Reed-Solomon code allows recovery of the message $\sigma$ from the code word $E(\sigma)$. Furthermore, for any messages $\sigma \neq \tau$,
$f_\sigma$ and $f_\tau$ must be distinct, so $E(\sigma)$ and $E(\tau)$ can agree at most $n - 1$ values. The code words $E(\sigma)$ and $E(\tau)$ have $m$ symbols, at most $n - 1$ of which are identical, so therefore $E$ is a code of distance $m - n + 1$. \hfill \Box