Recap - Zero Knowledge, Mignotte's Secret Sharing Schemes
What have seen? What are we discussing today? What is coming later?

Lecture 9
- Secure Multi Party Computation
- Shamir’s Secret Sharing Schemes

Lecture 10
- Recap on Zero-Knowledge (Identification) protocols
- The Schnorr protocol
- Recap on the Chinese Remainder Theorem
- Mignotte’s Secret Sharing Scheme
- Exercise

Lecture 11
- Hash Functions
- Collision Resistance
- The birthday paradox
Reminder: Victor the Verifier & Peggy the Prover

Goal of V
Accept a prover $P$, only if $P$ provides a true statement!

Goal of $P$
Prove a statement to $V$ i.e., convince that the statement is true
Examples of statements:
“I am Peggy” (for identification) or
“I have the secret key for this public key”
Reminder: Zero-Knowledge Protocols

I am convinced that she has the secret key but I don't know what the key is

I have the secret key

Zero-Knowledge (intuition): A honest P can convince V of the validity of a statement without revealing any information beyond the truth of the statement.

What does this mean?

- No matter what the verifier does, the verifier will not extract any useful information from the prover (e.g. the secret key) while, at the same time, the verifier will be convinced that the prover knows the secret connected to the statement.

Attention: The verifier should not be able to transfer this knowledge i.e., Victor (the verifier) cannot convince anyone else that Peggy has the secret key!

How is this possible? The magic of Crypto!
The Schnorr Identification Protocol

Group $G$ of prime order $q$ with generator $g$.

Choose a random value $c \in \{1, 2, \ldots, q - 1\}$

Check

$R \overset{?}{=} g^s \cdot X^{-c}$

$= g^{r+cx \mod q} \cdot (g^x)^{-c}$

$= g^{r+cx-cx \mod q}$

$= g^r = R$

Schnorr protocol in practice

- **Peggy** (the prover $P$) can precompute a number of pairs $(r, R)$.
- Then, her only computation during the protocol execution is $r + cx \mod q$.
- Useful if $P$ is a smart card with limited computational power
The Schnorr protocol: Basic properties

**Completeness:** The protocol is complete since any valid prover will have the verifier to accept him with probability 1.

**Soundness:** An adversary who does not know the secret key $x$ thus has to guess the response $s$ with a probability of $\frac{1}{q}$. Thus, the soundness is $\frac{1}{q}$ of being correct.

The security of Schnorr identification protocol relies on the discrete log problem.

**Zero-Knowledge:** Victor (the verifier) does not find out the secret key but is convinced that Peggy (the prover) has it.

The security of Schnorr identification protocol relies on the discrete log problem.
Quiz Question

If we repeat the Schnorr protocol $n$ times what is the probability that a dishonest prover will convince the verifier?

**Hint:** A dishonest prover who does not know $x$ has probability $1/q$ to convince Victor that she knows the secret key (password).

Dishonest, means that he does not have the secret information $s$ and he claims that he has!

**Answer:** $(\frac{1}{q})^n$
Quiz Question

Can Victor using the exchanged messages of the protocol \((R, c, s)\) to persuade someone else that Peggy has the secret \(x\)?

Answer: No! \(V\) could have produced \((R, c, s)\) by generating \(c\) and \(s\) at random and computing \(R = g^sX^{-c}\)
Quiz Question

Go to: http://socrative.com/
or scan the QR code
Student Login
Classroom: CRYPTOCHALMERS

Quiz question:
Consider that a prover \( P \) (Peggy) runs the Schnorr protocol twice with the same commitment (i.e., value \( R \)). This means that the transcripts of the two protocol runs are: \((R, c, s)\) and \((R, c', s')\). Can an eavesdropper recover the secret key \( x \)?

A No, it is a zero-knowledge protocol so no-one can recover the secret key.

B Yes, by using the equation: \( R = g^s \cdot X^{-c} = g^{s'} \cdot X^{-c'} \). ⇐

C It depends on the size of \( x \).

Indeed it holds:
\[
R = g^s \cdot X^{-c} = g^{s'} \cdot X^{-c'} \iff g^s \cdot (g^x)^{-c} = g^{s'} \cdot (g^x')^{-c'} \\
s + x(-c) = s' + x(-c') \mod q \iff \iff x = \frac{s-s'}{c-c'} \mod q
\]

Remember: This property applies to all \( \Sigma \) protocols and is known as special soundness!
Reminder: The Chinese Remainder Theorem (CRT)

Chinese Remainder Theorem (simplified version)
Let $p$ and $q$ be distinct primes and $N = p \cdot q$. Let $a \in \mathbb{Z}_p$ and $b \in \mathbb{Z}_q$.

Then the system of congruences:

$$
\begin{align*}
  x &\equiv a \pmod{p} \\
  x &\equiv b \pmod{q}
\end{align*}
$$

is always solvable.

**How do we compute $x$?**
If $s, r \in \mathbb{Z}$ are two integers satisfying $sp + rq = 1$ (Bézout identity) (i.e., $s$ and $r$ are computed using the Extended Euclidean Algorithm (EEA)), to find $x$ we compute: $x = a \cdot r \cdot q + b \cdot s \cdot p$.

**NOTE:** ‘The’ solution $x$ is **unique** $\pmod{N}$
At Alice's birthday party there will be either 7 or 11 guests. How many slices of cake shall Alice cut in order to be sure that in the first case there will be just 1 slice left, and in the second case only 3?

Solution:

Using EEA we have:
\[ 2 \cdot 11 + (-3) \cdot 7 = 22 - 21 = 1 \]

By the CRT we have:
\[ x = 1 \cdot 22 + 3 \cdot (-21) \]
\[ = 22 - 63 = -41 = 36 \mod 77 \]

EEA stands for Extended Euclidean Algorithm
CRT stands for Chinese Remainder Theorem
Reminder: (Threshold) Secret Sharing Schemes (SSS)

Secret Sharing Schemes

A secret-sharing scheme usually involves:

- a dealer \( D \) who has a secret \( s \)
- \( n \) parties \( P_1, \ldots, P_n \)

A secret-sharing scheme is a method that be used so that the dealer distributes shares of the secret \( s \) to the \( n \) parties in such a way that:

1. any subset of \( t + 1 \) parties can reconstruct the secret from its shares and
2. any subset of \( t \) parties cannot retrieve any partial information on the secret \( s \).

My secret is \( s \)!
Mignotte’s threshold secret sharing scheme

Useful for the exercises/exams!!

A Secret Sharing Scheme based on the Chinese Remainder Theorem (CRT)

[Image of a red heart with a smiley face and a heart shape]

Remember: Threshold is the number $t$ of maximum corrupted parties tolerated by the scheme (i.e., any subset of $t$ parties cannot reconstruct the secret $s$).

- **STEP 1:** Choose two integers $n$ (number of parties), $t$ (threshold), such that $t \leq t \leq n - 1$.

Let $m_1, m_2, \ldots, m_n$ be a Mignotte sequence

i.e., $m_1 < m_2 < \ldots < m_n$ are $n$ positive integers, pairwise co-prime,

satisfying $m_1 \cdot m_2 \cdot \ldots \cdot m_{t+1} > m_{n-t+1} \cdot m_{n-t+2} \cdot \ldots \cdot m_n$

What does co-prime mean?

$GCD(m_i, m_j) = 1$ for all $i \neq j$, and $i, j \in \{1, 2, \ldots, n\}$

- **STEP 2:** Let $s$ be the secret that the **Dealer** wants to share with the $n$ parties

($s$ must satisfy $m_1 \cdot \ldots \cdot m_{t+1} > s > m_{n-t+1} \cdot \ldots \cdot m_n$)

The Dealer computes $s_i \equiv s \pmod{m_i}$ and gives $s_i$ to party $P_i$
Mignotte’s threshold secret sharing scheme

Useful for the exercises/exams!!

**A Secret Sharing Scheme based on the Chinese Remainder Theorem (CRT)**

Any $t + 1$ different parties can compute the solution $x$

- **Step 3:** (secret recovery)
  
  $$
  \begin{align*}
  x &\equiv s_{i_1} \pmod{m_{i_1}} \\
  x &\equiv s_{i_2} \pmod{m_{i_2}} \\
  \vdots \\
  x &\equiv s_{i_{t+1}} \pmod{m_{i_{t+1}}}
  \end{align*}
  $$

By the **Chinese Remainder Theorem** this system has a unique solution modulo $m_{i_1} \cdot m_{i_2} \cdot \ldots \cdot m_{i_{t+1}}$ and by the properties of the Mignotte sequence, this solution is the secret, i.e., $x = s$. Any set of $t$ or less parties will not be able to find $s$. 
Mignotte’s Secret Sharing Scheme: numerical example

Let $n = 4$ and $t = 2$, $m_1 = 3$, $m_2 = 4$, $m_3 = 5$, $m_4 = 7$

- **Step 1**: Check that the given values form a Mignotte series
  
  $GCD(m_i, m_j) = 1$? YES
  
  $m_1 < m_2 < m_3 < m_4$? $3 < 4 < 5 < 7$ YES!
  
  $m_1 \cdot m_2 \cdot m_3 > m_3 \cdot m_4$? $3 \cdot 4 \cdot 5 = 60 > 35 = 7 \cdot 5$ YES!

- **Step 2**: Let $s = 40$ be the secret to be shared.
  
  Check that the given value is in the Mignotte range.
  
  Compute the shares $s_i = s \pmod{m_i}$

  $s_1 = 40 \pmod{3} = 1 \pmod{3}$
  
  $s_2 = 40 \pmod{4} = 0 \pmod{4}$
  
  $s_3 = 40 \pmod{5} = 0 \pmod{5}$
  
  $s_4 = 40 \pmod{7} = 5 \pmod{7}$
Mignotte’s Secret Sharing Scheme: numerical example

Let \( n = 4 \) and \( t = 2 \), \( m_1 = 3 \), \( m_2 = 4 \), \( m_3 = 5 \), \( m_4 = 7 \)

The secret shares are:

\[ s_1 = 1 \pmod{3}, \ s_3 = 0 \pmod{5} \]
\[ s_2 = 0 \pmod{4}, \ s_4 = 5 \pmod{7} \]

\[ \text{STEP 3 (secret recovery)} \]

\[
\begin{align*}
    x &= s_2 \pmod{m_2} = 0 \pmod{4} \\
    x &= s_3 \pmod{m_3} = 0 \pmod{5} \\
    x &= s_4 \pmod{m_4} = 5 \pmod{7}
\end{align*}
\]

Applying the Chinese Remainder Theorem

\[ ps + qt = GCD(p, q) \]

\[ 5(1) + 4(-1) = 1, \text{ EEA between 4 and 5} \]

So from the CRT on the first two equations we have:

\[ x = 0 \cdot 5 + 0 \cdot (-4) = 0 \pmod{20} \]

\[ 20(-1) + 7(3) = 1, \text{ EEA between 20 and 7} \]

\[ s = x = 5 \cdot (-20) + 0 \cdot 21 = -100 \pmod{140} = 40 \pmod{140} \]

Yes!! The recovered secret \( s = x = 40 \) is correct!!
Overview - What have we covered in Cryptographic Protocols?

output: 
\[ y = f(x_1, x_2, x_3, x_4, x_5, x_6) \]

**SMPC**

**Attacks**
- Known Key Attack
- Reflection Attack
- Man-in-the-Middle Attack

**Protocols**
- Key-Exchange Protocols (Needham Schroeder, Diffie-Hellman)
- Identification (Sigma) Protocols (Fiat-Shamir, Schnorr)

**Advanced Schemes**
- Shamir Secret Sharing
- Mignotte’s Secret Sharing
- Addition protocol SMPC

**Properties**
- Zero-Knowledge
- Privacy
- Completeness
- (special) Soundness
- Correctness

SMPC stands for Secure Multi-Party Computation.
5. Assume that we have three parties $P_1, P_2$ and $P_3$ and that we tolerate $t = 1$ corrupted party. Assume that we work in $\mathbb{Z}_{11}$ and each of the parties have a secret value $a = 2$, $b = 4$ and $c = 1$ correspondingly. The three parties want to compute the sum $\sigma = a + b + c$ while keeping their corresponding value secret. Using Shamir’s secret sharing show how to calculate the sharing of $a, b, c$ and of their sum $\sigma$.

More precisely if we denote by $a_1, a_2, a_3$ the shares of the secret value $a$ and we denote similarly the shares of $b, c$ and $\sigma$. Then:

(a) Fill in the following table: (5 p)

<table>
<thead>
<tr>
<th></th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = 2$</td>
<td>$a_1$</td>
<td>$a_2$</td>
<td>$a_3$</td>
</tr>
<tr>
<td>$b = 4$</td>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$b_3$</td>
</tr>
<tr>
<td>$c = 1$</td>
<td>$c_1$</td>
<td>$c_2$</td>
<td>$c_3$</td>
</tr>
<tr>
<td>$\sigma = a + b + c$</td>
<td>$\sigma_1$</td>
<td>$\sigma_2$</td>
<td>$\sigma_3$</td>
</tr>
</tbody>
</table>

(b) We want that $P_1$ only learns $\sigma = 7$ and nothing about $b$ and $c$. Show that if $P_1$ makes the hypothesis that $b = 3$ and $c = 2$ then this cannot be excluded from the possible solutions. $P_1$ has the following view:

<table>
<thead>
<tr>
<th></th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = 2$</td>
<td>$a_1$</td>
<td>$a_2$</td>
<td>$a_3$</td>
</tr>
<tr>
<td>assumes $b = 3$</td>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$b_3$</td>
</tr>
<tr>
<td>assumes $c = 2$</td>
<td>$c_1$</td>
<td>$c_2$</td>
<td>$c_3$</td>
</tr>
<tr>
<td>$\sigma = a + b + c$</td>
<td>$\sigma_1$</td>
<td>$\sigma_2$</td>
<td>$\sigma_3$</td>
</tr>
</tbody>
</table>

where the blue values (i.e., $a, a_1, a_2, a_3, b_1, c_1, \sigma, \sigma_1, \sigma_2, \sigma_3$) denote the values that $P_1$ already knows (you have them if you solve question (a)) This means that $P_1$ does not know only the values: $b, c, b_2, b_3, c_2, c_3$. (5 p)
Things to Remember

- Describe how the Schnorr identification protocol works.
- If a dishonest prover that does not have the secret can predict the verifier’s challenges can he succeed in running the Schnorr protocol?
- What is special soundness?
- How does the Mignotte’s secret sharing scheme work?
References:

- Secure Multiparty Computation and Secret Sharing: An Information Theoretic Approach, R. Cramer, I. Damgard, J. Buus Nilsen (Chapters 1, 3.1, 3.2, 3.3.4)
- Additional Material: Handbook of Applied Cryptography?, Menezes, Oorschot, Vanstone (Chapters 10.3, 10.3.1, 10.3.2, 10.4, 10.4.1, 10.4.3, 10.4.4) http://cacr.uwaterloo.ca/hac/about/chap10.pdf

Thank you for your attention!