Today’s Agenda

- Wrap Up: Cramer-Shoup Cryptosystem

1 Wrap Up: Cramer-Shoup Cryptosystem

Let us recall the Reduced Cramer-Shoup cryptosystem:

Key Generation: We have setup $G$ of order $q$ and generators $g, h \leftarrow G$. Output $SK = x, y, a, b$, and the public key will be $PK = (A, B)$, where $A = g^x h^y$ and $B = g^a h^b$.

Encryption: On input $m \in G$, choose $r \leftarrow \mathbb{Z}_q$, output ciphertext $c = (R, S, P, T) = (g^r, h^r, A^r m, B^r)$.

Decryption: On input $c = (R, S, P, T)$:

1. Check if $T = R^a S^b$, output $\perp$ if not
2. Output $m = P/R^x S^y$

For our security we wish to show that $\forall$ PPT $A$, it can’t distinguish between the following two experiments:

<table>
<thead>
<tr>
<th>Real</th>
<th>Simulated</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G, g, h$ setup correctly</td>
<td>$G, g, h$ setup correctly</td>
</tr>
<tr>
<td>$(pk, sk) \leftarrow \text{KeyGen}(G, g, h)$</td>
<td>$(pk, sk) \leftarrow \text{KeyGen}(G, g, h)$</td>
</tr>
<tr>
<td>$(info, m) \leftarrow A^{\text{Dec}(\cdot)}(pk)$</td>
<td>$(info, m) \leftarrow A^{\text{Dec}(\cdot)}(pk)$</td>
</tr>
<tr>
<td>$c \leftarrow \text{Enc}(pk, m)$</td>
<td>$c \leftarrow \text{Sim}(pk)$</td>
</tr>
<tr>
<td>$b \leftarrow A(info, c)$</td>
<td>$b \leftarrow A(info, c)$</td>
</tr>
</tbody>
</table>

We want that there exists a negl. function $\nu()$:

$$| \Pr[b = 0 | A \text{ is in the real experiment}] - \Pr[b = 0 | A \text{ is in the simulated experiment}] | = \nu(k)$$

A possible algorithm for the simulator: On input $pk$, the simulator outputs completely random elements $(R, S, P, T) = (g^r, g^r, g^r, g^r)$. Now we will show a reduction who is trying to break the DDH assumption. It receives a tuple and it has to decide whether it is a DDH tuple or random.

On input $(G, q, g_1, g_2, g_3, g_4)$, reduction does the following:

1. Generate $(sk, pk)$ honestly with $g = g_1$ and $h = g_2$
2. Invoke $A(pk)$ and answer its decryption queries
3. When $A$ outputs $(info, m)$, form ciphertext $(R, S, P, T)$
4. Choose message $m$.
• Suppose input tuple \((G, g_1, g_2, g_3, g_4)\) is the well-formed tuple \((G, g, g^a, g^\beta, g^{a\beta})\). Form the challenge ciphertext \(c\) as

\[
(R = g_3, S = g_4, P = A' m = R^x S^y m, T = B^r = R^a S^b)
\]

• Suppose input tuple \((G, g_1, g_2, g_3, g_4)\) is \((G, g, g^a, g^\beta, g^\gamma)\) for a random \(\gamma\). In this case we want our ciphertext to be distributed exactly as the one that our simulator will produce. Form the challenge ciphertext \(c\) as

\[
(R_1 = g^{r_1} = g_3, R_2 = g^{r_2} = g_4 = g^\gamma = g^{a\beta + \delta}), S = A' m = R^1_1 R^2_2 g^a = g^{\beta x + \gamma y + \mu}, T = B^r = R^1_1 R^2_2)
\]

5. Invoke \(A(\text{info}, (R, S, P, T))\) and output what it outputs.

**Definition 1.** A ciphertext \((R, S, P, T)\) where \(R = g^{r_1}, S = h^{r_2}\) with \(r_1 \neq r_2\) and \(T = R^a S^b\) is a decryptable invalid ciphertext.

No we claim the following things about the reduction:

1. If input tuple is DDH then the reduction is real: The steps of the reduction are equivalent to the steps in the real experiment and also if we have a DDH tuple then the \(S, P, T\) values will also work out correctly as they would be in the real experiment.

2. If input tuple is random then the reduction is simulated: No \(A\) even with an unbounded runtime (but poly. number of queries) can form a decryptable invalid ciphertext with non-negl. probability. **Proof:** To prove this let us look at the following mental experiments through hybrids:

• **Green Process:** Instead of knowing the complete \(sk = (x, y, a, b)\), suppose you only know \((s, t)\) such that \(A = g^s\) and \(B = g^t\). Let us say we also know \(\alpha\) such that \(h = g^\alpha\). Also \((R, S, P, T)\) is picked at random. So now we will also modify how decryption works. Instead of our check whether \(T = R^a S^b\) we will now check whether \(R^a = S\) and \(R^t = T\) and if the check goes through output the decryption as \(P/R^a\)

• **Red Green Process:** Instead of knowing the complete \(sk = (x, y, a, b)\), suppose you only know \((s = x + ay, t = a + \alpha b)\) such that \(A = g^s\) and \(B = g^t\). Also do both the checks now: check whether \(T = R^a S^b\) and also whether \(R^a = S\) and \(R^t = T\) and if the check goes through output the decryption as \(P/R^a\)

• **Red Process:** Again we have that \((s = x + ay, t = a + \alpha b)\) such that \(A = g^s\) and \(B = g^t\). Now put \((R = g_3 = g^\beta, S = g_4 = g^\gamma, P = R^a S^b m = g^{\beta x + \gamma y + \mu})\) where \(m = g^\mu\). With this we have that provided the reduction never has to decrypt a decryptable invalid ciphertext if the input tuple is random then the reduction is equivalent to simulated experiment.

Hence proved that the reduced Cramer-Shoup cryptosystem is CCA-1 secure. Now we will prove security for the full version of Cramer-Shoup:

**Key Generation:** We have setup \(\mathbb{G}\) of order \(q\) and generators \(g, h \leftarrow \mathbb{G}\). Output \(SK = x, y, a, b, w, z\), and the public key will be \(PK = (A, B, C)\), where \(A = g^x h^y\) and \(B = g^a h^b\) and \(C = g^w h^z\).

**Encryption:** On input \(m \in \mathbb{G}\), choose \(r \leftarrow \mathbb{Z}_q\), output ciphertext \(c = (R, S, P, T) = (g^r, h^r, A' m, (BC^H(R, S, P))^r)\) where \(H\) is a CRHF.

**Decryption:** On input \(c = (R, S, P, T)\):

1. Check if \(T = R^{aH(R, S, P)w} S^bH(R, S, P)^z\), output \(\perp\) if not
2. Output $m = P/R^xS^y$

The proof outline is similar to that of the reduced cryptosystem. Again we define the different processes: red, red-green, green and we can prove that the view of the adversary is same in all the processes. Note that getting CCA-2 security is very tricky and the analysis has to be done very rigorously. So even for the most basic primitive in cryptography which is encryption it is a challenge to prove the highest notion of security for it.