1 General

1.1 Review of Last Week

- The ElGamal encryption scheme is an encryption scheme in the DL setting, and its security is based on the hardness of the Decisional Diffie-Hellman (DDH) problem.

\[
G = \{g^0, g^1, \ldots, g^{n-1}\}
\]

\[
\text{Secret Key (SK)} \leftarrow x \in \mathbb{Z}_m
\]

\[
\text{PrivateKey (PK)} \leftarrow y = g^x
\]

\[
\text{Enc}_y(m) : r \leftarrow \mathbb{Z}_m; \ k = g^r; \ c = my^r; \ \text{Output}(k, c)
\]

\[
\text{Dec}_x(k, c) : m \leftarrow ck^{-x}
\]

- If DDH is secure then ElGamal is also secure w.r.t IND-CPA

- RSA

\[
\phi(N) = (p-1)(q-1)
\]

\[
d = e^{-1} \mod \phi(N)
\]

\[
x \in \mathbb{Z}_N
\]

\[
y = x^e \mod N
\]
2 ElGamal Encryption

**Proposition:** ElGamal is not IND-CCA2

**Generic Construction** for IND-CCA2 asymmetric key encryption: We can convert any IND-CPA encryption scheme into IND-CCA2 by including a signature on the ciphertext \((k, c)\) value (using the private key). Alice can encrypt using Bob’s public key as before and add a signature on the ciphertext using her private key; on receiving side, Bob can verify the signature using Alice’s public key, and decrypt the ciphertext using his own private key.
3 Cramer-Shoup Encryption

The Cramer-Shoup system is an asymmetric key encryption algorithm, and is proven to be secure against adaptive chosen ciphertext attack (IND-CCA2) using standard cryptographic assumptions. Developed by Ronald Cramer and Victor Shoup in 1998, it is an extension of the ElGamal cryptosystem. Collision-resistant hash function and additional computations are used, thus making the ciphertext twice as large as in ElGamal.

\[ g_1, g_2 \in G_q \]
\[ (SK) \rightarrow (x_1, x_2, y_1, y_2, z) \in \mathbb{Z}_q \]
\[ (PK) \rightarrow \begin{cases} 
  c = g_1^{x_1} g_2^{x_2} \\
  d = g_1^{y_1} g_2^{y_2} \\
  h = y_1 
\end{cases} \]

\[
\text{Enc}(m): \\
\begin{align*}
  r & \xleftarrow{\$} \mathbb{Z}_q \\
  u_1 &= g_1^r \\
  u_2 &= g_2^r \\
  e &= mh^r \\
  \alpha &= H(u_1, u_2, e)
\end{align*}
\]

(Where \( H \) is a collision resistant hash function)

\[ v = e^{\alpha d^{\alpha}} \]

The result is \((u_1, u_2, e, v)\)

\[
\alpha = H(u_1, u_2, e) \\
\text{Verify that } e^{d \alpha} = u_1^{\alpha} u_2^{\alpha} (u_1^{y_1} u_2^{y_2})^{\alpha} = v \]

If verified, compute: \( m = e u_1^{-z} \)

**Theorem:** If \( H \) is CR2 and DDH is hard in \( G_q \), then the Cramer-Shoup algorithm is IND-CCA2.

The system is IND-CCA2 secure if DDH assumption holds in \( G \), and \( H \) is collision resistant.

\[ \alpha' = H(u_1', u_2', e') \]
4 RSA Encryption

We will begin by describing textbook RSA, named after Rivest, Shamir, and Adleman. RSA is built on the group \( (\mathbb{Z}_N^*) \) where \( N = pq \) such that \( p \) and \( q \) are primes. We also compute the Euler’s toitent \( \phi(N) = (p - 1)(q - 1) \), and find an integer \( e \) such that \( \gcd(\phi(N), e) = 1 \). Now we compute \( d = e^{-1} \mod \phi(N) \)

Finally, we are left with the secret and public keys:

\[
(SK) \rightarrow (d, p, q) \\
(PK) \rightarrow (N, e)
\]

\[
Enc_{N,e}(m \in \mathbb{Z}_n^*) : c = m^e \mod N
\]

\[
Dec_d(c) : m = c^d \mod N
\]

We see that textbook RSA is not randomized, therefore it is not IND-CPA. We can also see that if \( c = Enc(m) \), then \( Dec(2^e c) = 2m \), and therefore textbook RSA is also not IND-CCA2 either.

**Proposition:** RSA is not IND-CPA

Textbook RSA is not randomized. And, so we can query the encryption oracle (i.e., encrypt ourselves) one of the messages and compare with the challenged ciphertext.

**Proposition:** RSA is not IND-CCA2

Adversary have access to decryption oracle. We have to use a Decryption oracle to generate a query. So we will massage the cipher text and will send \( c' = (M)^e \mod N \).
(we pick $M$ by ourself.)

So, RSA is not CCA-2 Secure.

4.1 RSA OAEP

RSA OAEP (Optimal Asymmetric Encryption Padding) is essentially a randomized variation of the RSA framework.

Key Generation: OAEP will require the same initial RSA parameters and keys as textbook RSA, so these are generated as usual. In addition, we will have the following parameters:

$$k = |N|$$

$$k_0, k_1 \in \mathbb{N} \text{ such that } k_0 + k_1 < k$$

$$n = k - (k_0 + k_1)$$

Messages will be of length $n$

$$G : \{0, 1\}^{k_0} \rightarrow \{0, 1\}^{k_1 + n}$$

$$H : \{0, 1\}^{k_1 + n} \rightarrow \{0, 1\}^{k_0}$$

(Where $H$ and $G$ are random functions)

$\text{Enc}(m)$: $m$ is a message of length $n$

$$r \xleftarrow{\$} \{0, 1\}^{k_0}$$

$$S = G(r) \mathbin{\|} (m||r^{k_1})$$

(Where $\|$ represents padding, this has a length of $n + k_1$)
4.2 OAEP History

1. First proposed in 1995 by Bellare and Rogaway. They proved that it is IND-CCA1, and it was thought that IND-CCA2 would be an eventual corollary of this work, however it was not proven secure.

2. In 2000, Victor Shoup called into doubt that the algorithm is IND-CCA2, but he proved that it is IND-CCA2 for the exponent \( e = 3 \), which is commonly used.

3. Fujisaki, Okamoto, Pointcheval showed that RSA-OAEP is secure against IND-CCA2.

4. In 2001, it is finally shown in full generality that RSA-OAEP satisfies the IND-CCA2 property, though this work indicates the following relation:

\[
\text{Adv}^{RSA-OAEP}(A) = q \cdot \text{ADV}^{RSA}(A)
\]

Where \( q \) is the number of calls to the random functions \( G \) and \( H \). This is an undesirable property because there may be many calls to these functions and this reduction does not offer exact security (and has security degradation by a factor of \( q \)). Some have suggested that a better proof of the security of OAEP may be possible.

5 Digital Signatures

KeyGen : (SK,PK)

Sign() : input \( \leftarrow \) m, SK

\( s \leftarrow \text{sign}_{SK}(m) \)

output \( (m,s) \)

Verify : \( \text{verify}_{pk}(m, s) = 1 \) or 0
5.1 Security Notion

In CMA attack, the adversary \(A\), has access to signature oracle and he is allowed to query any message of his choice and request the corresponding signature. After requesting a number of signatures, the adversary attempts to come up with a forgery – i.e., a new message and the corresponding signature. If he can create a valid forgery, he is said to have won the CMA game. Note that there is no need to have the Verification oracle (unlike the case of MACs) because the adversary can himself verify the validity of the forged signature by using the public key of the signer.

Construction : RSA Signature

KeyGen : \((N, e), d\)

\(\text{Sign}_d(m) : S = m^d \mod N\)

\(\text{Verify}_{N, e}(m, S) = S^e = m \mod n\)

Proposition: Textbook RSA in not CMA secure

Textbook RSA is homomorphic. In other words, if we multiply the signatures on two different messages \(m_1\) and \(m_2\), we can get a corresponding signature on a new message \(m_1 \ast m_2\).

\[
(m_1m_2)(s_1s_2) \Rightarrow (m_1m_2, s_1s_2)
\]

\[
s_1 = m_1^d \mod N
\]

\[
s_2 = m_2^d \mod N
\]

\[
s_1s_2 = (m_1m_2)^d \mod N
\]

We use a randomized version of the textbook RSA signatures known as the Full Domain Hash (FDH) RSA signatures. It works as follows:

KeyGen : \((N, e) \Rightarrow p_K\)

\[d \Rightarrow s_K\]

Sign : \((H(m))^d \mod N\)

\((m, s)\)

Verify : \(S^e = H(m) \mod N\)

Note that now the homomorphic property does not hold (as shown below) and FDH RSA signatures remain secure under the RSA assumption given that \(H()\) behaves as a random function.

\[
s_1 = (H(m_1))^d \mod N
\]

\[
s_2 = (H(m_2))^d \mod N
\]

\[
s_1 s_2 = [H(m_1) \cdot H(m_2)]^d \mod N
\]

\[
\neq [H(m_1 \cdot m_2)]^d \mod N
\]
6 Certification of Public key

With asymmetric encryption, we need a method of binding a public key with an identity, so that an attacker does not intercept a transmission of a public key and replace the key with the attacker's public key. The central mechanism in the verification of public keys is the Trusted Third Party (TTP).

Certification is a process using which a trusted third party (known as certification authority or CA) binds the identity of the bearer of the certification with his/her public key. In particular, the CA signs (ID,PK) which essentially becomes a certificate. By verifying the signature of the CA on this certificate, any entity can be sure that PK is the public key of corresponding to a person with identity ID, as long as the CA is trusted.