University of Illinois, Urbana Champaign
CS 598DK Special Topics in Cryptography

Instructor: Dakshita Khurana
Scribe: Aniket Murhekar, Rucha Kulkarni
Date: September 11, 2019

LECTURE
5

Private-Key and Public-Key Encryption

We discussed pseudo-random functions in the last lecture. In this lecture, we focus on encryption schemes. Today we will cover the following topics.

- Define private-key and public-key encryption schemes and understand what it means for them to be secure.
- Define and distinguish between single-message and multi-message encryption schemes.
- Construct a private-key encryption scheme using pseudo-random functions (PRFs) and present a proof of its security.
- Show that a secure single-message public key encryption scheme is also a secure multi-message encryption scheme.
- Describe a public key encryption scheme, El-Gamal.

5.1 Recap: Pseudo-random functions

Recall from the last lecture that a pseudo-random function is a family of functions which, intuitively speaking, behaves like a randomly chosen function would behave.

More concretely, we say that a function family $F : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^m$ is pseudo-random if no efficient adversary $\mathcal{A}$ can distinguish the function $F(s,\cdot)$ (for a uniformly chosen s) from a function chosen uniformly at random from the set $\mathcal{F}$ of all functions from $\{0,1\}^l$ to $\{0,1\}^m$. Here our adversaries are non-uniform probabilistic polynomial-time Turing machines with oracle access to either $F(s,\cdot)$ or a function in $\mathcal{F}$. Thus,

Definition 5.1. *Pseudo-random function (PRF).* A family of functions $F : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^m$ is said to be pseudo-random if for all non-uniform probabilistic polynomial-time oracle Turing machines $\mathcal{A}$, we have that:

$$|\Pr[\mathcal{A}^{F(s,\cdot)}(1^k) = 1] - \Pr[\mathcal{A}^f(1^k) = 1]| \leq \text{negl}(k)$$

where the first probability is taken over the uniform choice of $s \in \{0,1\}^k$ and the randomness of $\mathcal{A}$, and the second probability is taken over the uniform choice of $f \in \mathcal{F}$ and the randomness of $\mathcal{A}$.

5.2 Encryption schemes and their security

We first define an encryption scheme. Informally, an encryption scheme should do the following: generate a key (or a pair of keys), encrypt a plaintext message to produce a ciphertext, and then decrypt a ciphertext using the key to recover the plaintext. Formally:

DEFINITION 5.2. *Encryption scheme.* An encryption scheme is a tuple of probabilistic polynomial-time (PPT) algorithms $(KeyGen, Enc, Dec)$:

1. $KeyGen$: On input 1^k, output two strings (ek, dk), using r_{keygen} as the randomness. Here ek is the encryption key or public key and dk is the decryption or private key.
2. Enc : On input encryption key ek and plaintext m, output the ciphertext c, using r_{enc} as the randomness.
3. Dec : On input decryption key dk and ciphertext c, output the plaintext m.

Note that Dec has to be a deterministic algorithm. Also note that the randomness r_{enc} used in encryption is not necessarily given as input to Dec.

We also distinguish between private-key and public-key encryption. In the former, the private key is shared between the two parties. We do not concern ourselves with how they exchanged this key. In the latter, the encryption is key is public, but the decryption is possible only with the knowledge of the private key.

We now discuss the properties that the algorithms should satisfy, namely, correctness and security.

Correctness. This requires that the algorithm Dec correctly decrypts the ciphertext of the message encrypted by the algorithm Enc:

$$\Pr[Dec(dk, Enc(ek, m)) = m] = 1 - \text{negl}(k), \tag{5.1}$$

where the probability is taken over the randomness used in the $KeyGen$ and Enc algorithms. The above notion is called *statistical correctness*. When the right-hand side is exactly 1, we say the scheme satisfies *perfect correctness*.

Security. When can we say that the scheme is secure? Suppose the scheme does not reveal the entire message m to adversaries who do not possess the decryption key. Can we call this scheme "secure"? Even if the scheme does not reveal m entirely, it could reveal parts of m, or maybe a function of the bits of m to adversaries, which itself is problematic and could reveal information about m. So our security definition needs to be strengthened. For this we consider a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

1. $\mathcal{A}$ is given 1^k and outputs two messages m_0 and m_1 of equal length.
2. $\mathcal{C}$ generates a key ek using $KeyGen$ and randomness r, and chooses a bit $b \in \{0, 1\}$ uniformly at random. She sends $c = Enc(ek, m_b)$ to $\mathcal{A}$.
3. $\mathcal{A}$ tries to guess b and outputs b'.
4. Set the variable $Game_{\mathcal{A}}$ to 1 if and only if $b = b'$.

We say that the scheme is secure if no non-uniform probabilistic polynomial-time (nuPPT) Turing machine can correctly guess b with probability much better than $1/2$, for every choice of m_0 and m_1. That is:

$$\forall \text{ nuPPT } \mathcal{A}, \forall m_0, m_1 \text{ s.t. } |m_0| = |m_1| = k : \Pr[Game_{\mathcal{A}} = 1] = \frac{1}{2} + \text{negl}(k). \tag{5.2}$$

A succinct way to put this is to say that the distributions $Enc(ek, m_0)$ and $Enc(ek, m_1)$ are *computationally indistinguishable* (a property hence forth also denoted by $\approx_c$) for all messages m_0 and m_1 of equal length. Note that we impose the condition of equal length, because the length of the message influences the length of the ciphertext.

Multi-message security. We extend the above definition of security to the setting when multiple messages are to be encrypted. We want our schemes to be secure against adversaries who could potentially look at encryptions of multiple plaintexts to decrypt a given ciphertext. For this, we define a game between an adversary $\mathcal{A}$ and challenger $\mathcal{C}$, for a given $p \leq \text{poly}(k)$, as follows:

1. $\mathcal{A}$ is given 1^k and outputs two tuples of p messages each: $M_0 = (m_{01}, m_{02}, \ldots, m_{0p})$ and $M_1 = (m_{11}, m_{12}, \ldots, m_{1p})$.
2. $\mathcal{C}$ generates a key ek using $KeyGen$ and randomness r_{keygen}, chooses a bit $b \in \{0, 1\}$ uniformly at random, encrypts all the messages in M_b using independent randomnesses $r_1, \ldots, r_p$, and sends across $C_b = (Enc(ek, m_{b1}; r_1), \ldots, Enc(ek, m_{bp}; r_p)))$ to $\mathcal{A}$.
3. $\mathcal{A}$ tries to guess b and outputs b'.
4. Set the variable $Game_{\mathcal{A},p}$ to 1 if and only if $b = b'$.

We say that the scheme is secure if no non-uniform probabilistic polynomial-time (nuPPT) Turing machine can correctly guess b with probability much better than $1/2$, for every pair of p-tuples M_0 and M_1, for every $p \leq \text{poly}(k)$. That is:

$$\forall \text{nuPPT}\mathcal{A}, \forall p \leq \text{poly}(k), \forall M_0, M_1 \text{ s.t. } |M_0| = |M_1| : \Pr[Game_{\mathcal{A},p} = 1] = \frac{1}{2} + \text{negl}(k).$$

Having formally defined what different types of security are, we now talk about the kind of public key encryption schemes, and the kind of private key encryption schemes, that will have both single and multi message security, and those that will only have one (single message) type.

In a public key setting, any adversary has access to the public key, hence can generate multiple ciphertexts of multiple messages by herself. If such a scheme is single message secure, then it cannot become insecure for multi message transfers. Else even for single messages, the adversary could generate encryptions of multiple messages, treat the single message's encryption as a part of a tuple of encryptions of multiple messages, use the algorithm that breaks (that is, decrypts with non-negligible probability) multi message security and decrypt with the same high probability the single message. This then contradicts single message security. Thus, by contrapositivity, single message security for public encryptions implies multi message security as well. This discussion is formalized in Section 5.4.

For private key encryption schemes however, guaranteeing multi message security requires additional properties than single message security, which we now discuss in the following section.

5.3 Secure private-key encryption scheme

Having understood the definition of encryption schemes and their security, we now look at examples. Recall the one-time pad (OTP) encryption scheme:

1. $KeyGen(1^k) = ek \xleftarrow{\$} \{0, 1\}^k$

2. $Enc(ek, m) = ek \oplus m$

3. $Dec(ek, c) = ek \oplus c$

It is easy to see that this satisfies both the properties of correctness and security when only one message is being encrypted. Thus in the single message setting, OTP is a perfectly secure encryption scheme. However, OTP suffers from a few drawbacks: the size of the key has to be equal to the size of the message, and the same key cannot be used to encrypt multiple messages. If the same key encrypts two messages m and m', then an adversary can compute $m \oplus m'$. Now an adversary can distinguish between the pairs $(0m_{01}, 0m_{02})$ and $(0m_{11}, 1m_{12})$ because the first bit of $c_1 \oplus c_2 = (0m_{b1} \oplus k) \oplus (bm_{b2} \oplus k))$ will be b, where $\in b\{0, 1\}$ is the choice of the challenger. Thus the adversary correctly guesses b with probability 1, rendering the OTP scheme insecure for multiple messages.

We now describe a secure private-key encryption scheme using PRFs. As we will show, this scheme is secure even in the multi-message setting. We first describe the scheme in the single message setting. Let $F : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^n$ be a PRF. Consider the scheme $\Pi = (KeyGen, Enc, Dec)$ given by:

1. $KeyGen(1^k) = s \xleftarrow{\$} \{0,1\}^k$, the seed of the PRF

2. $Enc(s, m; r) = \langle F(s, r) \oplus m, r\rangle$, where $r \xleftarrow{\$} \{0,1\}^l$, and $n = |m|$

3. $Dec(s, \langle c_1, c_2\rangle) = c_1 \oplus F(s, c_2)$

Correctness: The correctness condition of Equation (5.1) can be re-phrased using the notation used in defining the encryption scheme as:

$$Dec(dk, Enc(ek, m)) = Dec(s, \langle F(s, r) \oplus m, r\rangle) = (F(s, r) \oplus m) \oplus F(s, r) = m.$$

Thus, Equation (5.1) follows.
Security: In the following lemma, We will prove this scheme has single message security, and then as a corollary we establish how multi message security also follows.

LEMMA 5.3. *For all messages m_0 and m_1 of equal length, the distributions $Enc(ek, m_0)$ and $Enc(ek, m_1)$ are computationally indistinguishable. Equivalently, the encryption scheme defined above has single message security.*

Proof. For a uniformly randomly chosen string r', we have from the definition of PRFs the following computational indistinguishability relation.

$$Enc(ek = s, m_0; r) \equiv (m_0 \oplus F(s, r), r) \approx_c (m_0 \oplus r', r).$$

Now we can say $(m_0 \oplus r', r) \approx_c (r', r)$, for any r', r sampled uniformly at random. Thus we get,

$$(m_0 \oplus r', r) \approx_c (r', r) \approx_c (m_1 \oplus r', r) \approx_c (m_1 \oplus PRF(ek, r), r) \equiv Enc(ek, m_1; r),$$

and the lemma follows. ∎

COROLLARY 5.4. *The encryption scheme Π described above has multi message security.*

Applying the same proof idea to tuples of encryptions,

$$C_b = (Enc(ek, m_{b1}; r_1), Enc(ek, m_{b2}; r_2), \ldots, Enc(ek, m_{bp}; r_p)), \ for\ b \in \{0,1\},$$

of two distinct long messages $M_b = (m_{b1}, m_{b2}, \ldots, m_{bp})$, we show that for all $i \in [p]$, $Enc(ek, m_{bi}; r_i)$ is computationally indistinguishable from its corresponding part in the other message $Enc(ek, m_{(1-b)i}; r_i)$. Thus, the entire messages are also indistinguishable computationally, proving the corollary.

5.4 Secure public-key encryption scheme

We first prove the following theorem, which when combined with a single message secure public key encryption (PKE) scheme will result in a multi message secure PKE.

THEOREM 5.5. *Single message PKE $\implies$ Multi message PKE.*

Proof. Recall the informal discussion that established this theorem from Section 5.2. We formalize this now for two part messages.

Fix the messages as $M_0 = (m_{01}, m_{02})$ and $M_1 = (m_{11}, m_{12})$. We will prove their encryptions $D_b = (ek, Enc(ek, m_{b1}; r), Enc(ek, m_{b2}; r'))$ for $b \in \{0,1\}$ are computationally indistinguishable. By definition of multi-message security, the theorem then follows.

We define an intermediate distribution $D'_0 = (ek, Enc(ek, m_{01}; r), Enc(ek, m_{12}; r'))$ towards proving $D_0 \approx_c D_1$. We first prove by contradiction that $D_0 \approx_c D'_0$. Suppose then, that this claim is false, then there exists an adversary $\mathcal{A}$ such that

$$Pr[\mathcal{A}(ek, Enc(m_{01}), Enc(m_{02})) = 1] - Pr[\mathcal{A}(ek, Enc(m_{01}), Enc(m_{12})) = 1] = \frac{1}{poly(k)}.$$

We can then construct a machine, which is commonly termed as a *reduction*, denoted by B, that works as follows.

1. B first samples two messages m_{02}, m_{12} uniformly at random.
2. B then uses the public key ek, creates ciphertext ct and sends the messages $(ek, Enc(m_{i2}, ct))$, for $i \in \{0,1\}$.
3. It now runs the adversary's algorithm $\mathcal{A}(ek, Enc(m_{i2}), ct)$.

If $\mathcal{A}$ distinguishes between $(ek, Enc(m_{02}), ct)$ and $(ek, Enc(m_{12}), ct)$ with probability $1/poly(k)$, then B can distinguish between $(ek, Enc(m_{02}))$ and $(ek, Enc(m_{12}))$ with the same non-negligible probability, breaking single message security, a contradiction.

Similarly, we can prove $D'_0 \approx_c D_1$. Together with $D_0 \approx_c D'_0$ this establishes $D_0 \approx_c D_1$.

We can extend the proof to show the scheme has multi-message security for greater than 2 message parts, by creating the sequence of intermediate tuples

$$D_i = (ek, Enc(ek, m_{01}; r), \ldots, Enc(ek, m_{0x}; r), Enc(ek, m_{1(x+1)}; r), \ldots, Enc(ek, m_{1p}; r))$$

for all x in $[p]$, whose initial parts are ciphertexts of message parts of the first message, followed by ciphertexts of the second message. In a chain of relations we can prove they are all computationally indistinguishable, thus proving the theorem. □

In the final part of this lecture, we discuss a specific secure PKE scheme.

5.5 El Gamal: A special public key encryption scheme

We now look at a PKE scheme that assumes the hardness of deciphering certain properties of cyclic groups for proving secure encryption. The different algorithms of the scheme are as follows.

1. $KeyGen(1^k, r)$: A cyclic group $\mathcal{G}$ of order q is fixed. Then an arbitrary generator of $\mathcal{G}$, denoted by g, is picked. The algorithm then samples an element of the group $x \xleftarrow{\$} \mathbb{Z}_q$ uniformly at random, and considers this as the private key. Let $h = g^x \mod q$. The public key is the tuple $ek = (\mathcal{G}, q, g, h)$.

2. $Enc(ek, m, r')$: The encryption algorithm first picks $y \xleftarrow{\$} \mathbb{Z}_q$ using r', and creates the ciphertext consisting of two parts $c = (c_1, c_2) = (g^y, m \cdot h^y)$.

3. $Dec(x, c_1, c_2)$: The decryption algorithm outputs $c_2 \cdot (c_1^x)^{-1}$.

The correctness of the scheme is immediate since

$$c_2 \cdot (c_1^x)^{-1} = m \cdot h^y \cdot ((g^y)^x)^{-1} = m \cdot g^{xy} \cdot (g^{yx})^{-1} = m$$

Intuitively, the security of this protocol relies on the fact that given $h = g^x$ and g, the discrete logarithm x is hard to compute. The Decisional Diffie-Hellman assumption is that the distributions $(G, q, g, g^x, g^y, g^{xy})$ for x, y chosen uniformly at random, and (G, q, g, g^x, g^y, g^z) for x, y, z chosen uniformly at random are computationally indistinguishable. Under this assumption, an efficient adversary who does not know the secret key, i.e., who does not have knowledge of the discrete logarithm x , cannot distinguish between the encryption $m \cdot h^y = m \cdot g^{xy}$, from the string $m \cdot g^z$, where z is chosen uniformly at random, with non-negligible probability. A formal proof of security will construct a reduction (similar to the proof of Theorem 5.5) that breaks the Decisional Diffie Hellman assumption, and hence we skip it here.

In the **next lecture**, we will use these encryption schemes to design Oblivious Transfers, which are protocols to securely transfer messages between two parties while guaranteeing that both parties do not receive more information than what the sender intended.

Acknowledgement

These scribe notes were prepared by editing a light modification of the template designed by Alexander Sherstov.