Cryptography

Sugata Gangopadhyay

Department of Computer Science and Engineering
Indian Institute of Technology Roorkee. Roorkee - 247667 INDIA
`sugata.gangopadhyay@cs.iitr.ac.in`

Abstract. Notes from Douglas Stinson and Maura Paterson's book [4] and Hoffstein, Pipher, and Silverman [1], Katz and Lindell [2], Knospe [3].

Keywords: Cryptography.

1 Encryption Schemes

An encryption scheme transforms plaintexts into ciphertexts and conversely. The encryption function is parametrised by keys, and encryption is either deterministic or randomised. The symbols $=$ and $:=$ denote deterministic assignment, $\leftarrow$ for any type of assignment, and $\overset{\$}{\leftarrow}$ denotes randomised assignment.

Definition 1. *An encryption scheme or cryptosystem consists of*

1. *A plaintext space $\mathcal{M}$, the set of plaintext or clear-text messages.*
2. *A ciphertext space $\mathcal{C}$, the set of ciphertext messages.*
3. *A key space $\mathcal{K}$, the set of keys.*
4. *A randomised key generation algorithm $Gen(1^n)$ that takes the security parameter n in unary form as input and returns a key $k \in \mathcal{K}$.*
5. *An encryption algorithm $\mathcal{E} = \{\mathcal{E}_k \mid k \in \mathcal{K}\}$ which is possibly randomised. It takes a key and a plaintext message as input and returns the ciphertext or an error if the plaintext is invalid. We write $c = \mathcal{E}_k(m)$ or $c \leftarrow \mathcal{E}_k(m)$, and $c \overset{\$}{\leftarrow} \mathcal{E}_k(m)$ if the algorithm is randomised. The error output is denoted by $\perp$.*
6. *A deterministic decryption algorithm $\mathcal{D} = \{\mathcal{D}_k \mid k \in \mathcal{K}\}$ takes a key and a ciphertext message as input and returns the plaintext or an error symbol $\perp$ if the ciphertext is invalid. We write $m = \mathcal{D}_k(c)$ or $m \leftarrow \mathcal{D}_k(c)$.*

We require that all algorithms are polynomial with respect to the input size. Since Gen takes a unary string $1^n = 1 \dots 1$ of length n as input, the key generation algorithm is polynomial in n. The scheme provides correct decryption if for each $k \in \mathcal{K}$ and all plaintexts $m \in \mathcal{M}$ on has $\mathcal{D}_k(\mathcal{E}_k)) = m$. The plaintexts, ciphertexts, and keys are binary alphabets. That is

$$\mathcal{M}, \mathcal{C}, \mathcal{K} \text{ are subsets of } \{0,1\}^* = \bigcup_{n \in \mathbb{N}} \{0,1\}^n.$$

The one-time pad The one-time pad is an example of a simple and powerful fixed length encryption scheme. It used the binary alphabet, and the key length is equal to the message length. The security parameter n defines the length of plaintexts, ciphertexts and keys:

$$\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n.$$

The key generation algorithm $Gen(1^n)$ outputs a uniform key $k \overset{\$}{\leftarrow} \{0,1\}^n$. A key k of length n is used only for one message, $m \in \{0,1\}$. The encryption $\mathcal{E}_k$ and the decryption $\mathcal{D}_k$ are indentical and defined by a simple vectorial XOR operation:

$$c = \mathcal{E}_k(m) = m \oplus k, \qquad m = \mathcal{D}_k(c) = c \oplus k.$$

The Vigenère cipher The Vigenère cipher is a classical variable length scheme over the alphabet of letters.

$$\mathcal{M} = \mathcal{C} = \Sigma^* \text{ and } \mathcal{K} = \Sigma^n, \text{ where } \Sigma = \{A, B, \dots, Z\} \cong \mathbb{Z}_{26}.$$

$Gen(1^n)$ generates a uniform random key string $k \overset{\$}{\leftarrow} \Sigma^n$ of length n. For encryption and decryption the message and the ciphertext is split into blocks of length n, although the last block can be shorter. The encryption and decryption are as follows:

$$c = \mathcal{E}_k(m) = \mathcal{E}_k(m_1 \| m_2 \| \cdots) = (m_1 + k \| m_2 + k \| \cdots) \bmod 26,$$
$$m = \mathcal{D}_k(m) = \mathcal{D}_k(c_1 \| c_2 \| \cdots) = (c_1 - k \| c_2 - k \| \cdots) \bmod 26.$$

For $n = 1$, one obtains a monoalphabetic substitution cipher. For each $n > 1$ it is an example of a polyalphabetic substitution cipher.
Transposition ciphers A transposition cipher over an arbitrary alphabet encrypts a plaintext of length n by reordering the characters. Keys are given by a random permutation of $\{1, 2, \dots, n\}$. Bit permutations are transposition ciphers over the binary alphabet. We observe that the frequency of characters is preserved by transposition ciphers.

2 Perfect Secrecy

An encryption scheme is perfectly secret if for all plaintexts $m, m' \in \mathcal{M}$ and all ciphertexts $c \in \mathcal{C}$:

$$\Pr[\mathcal{E}_k(m) = c] = \Pr[\mathcal{E}_k(m') = c].$$

The probabilities are computed over randomly generated key $k \in \mathcal{K}$.

Lemma 1. *An encryption scheme is perfectly secret if and only if for every probability distribution over $\mathcal{M}$, every plaintext m and every ciphertext c for which $\Pr[c] > 0$, the probability of m and the conditional probability of m given c coincide:*

$$\Pr[m \mid c] = \Pr[m].$$

Proof. Suppose the encryption scheme is perfectly secret, therefore

$$\Pr[\mathcal{E}_k(m) = c] = \Pr[\mathcal{E}_k(m') = c].$$

This equation is equivalent to

$$\Pr[c \mid m] = \Pr[c \mid m'].$$

Let $\mathcal{M} = \{m_i \mid i = 0, 1, \ldots, |\mathcal{M}| - 1\}$. The condition of perfect secrecy means that

$$\Pr[c \mid m_0] = \Pr[c \mid m_1] = \cdots = \Pr[c \mid m_{|\mathcal{M}|-1}].$$

The total probability rule tells us

$$\Pr[c] = \sum_{i=0}^{|\mathcal{M}-1|} \Pr[c \mid m_i] \Pr[m_i].$$

Since all the conditionals are the same, for each integer $j \in \{0, 1, \ldots, |\mathcal{M}| - 1\}$ we have

$$\begin{aligned}
\Pr[c] &= \sum_{i=0}^{|\mathcal{M}-1|} \Pr[c \mid m_i] \Pr[m_i] \\
&= \sum_{i=0}^{|\mathcal{M}-1|} \Pr[c \mid m_j] \Pr[m_i] \\
&= \Pr[c \mid m_j] \sum_{i=0}^{|\mathcal{M}-1|} \Pr[m_i] \\
&= \Pr[c \mid m_j] \times 1 = \Pr[c \mid m_j].
\end{aligned}$$

This means that for all $m \in \mathcal{M}$ and $c \in \mathcal{C}$

$$\Pr[c] = \Pr[c \mid m].$$

On the other hand, if $\Pr[c] = \Pr[c \mid m]$, then for any two messages $m, m' \in \mathcal{M}$ and any $c \in \mathcal{C}$, we have $\Pr[c \mid m] = \Pr[c \mid m']$. The last equation implies

$$\Pr[\mathcal{E}_k(m) = c] = \Pr[\mathcal{E}_k(m') = c]$$

which is the condition for perfect secrecy.

$$\begin{aligned}
\Pr[m \mid c] &= \frac{\Pr[c \mid m] \Pr[m]}{\Pr[c]} \\
&= \Pr[m].
\end{aligned}$$

We also note that $\Pr[m \mid c] = \Pr[m]$ for all $m \in \mathcal{M}$ and $c \in \mathcal{C}$ implies that $\Pr[c \mid m] = \Pr[c]$. Thus, we have proved the equivalence of two definitions of perfect secrecy. $\square$

Theorem 1. *The one-time pad is perfectly secure if the key is generated by a random bit generator and is only used once.*

Proof. Let n be the security parameter of the one-time pad. Suppose m_0, m_1 are plaintexts and c is a ciphertext of length n. Then there is exactly one key k_0 of length n which encrypts m_0 into c and in fact $k_0 = m_0 \oplus c$. Since we have assumed an uniform distribution of keys, we have $\Pr[\mathcal{E}_k(m_0) = c] = \frac{1}{2^n}$. The same holds for m_1, which proves the theorem. □

Example 1. Suppose a Vigenère cipher of key length 3 is used to encrypt four characters. Let $c = (y_1, y_2, y_3, y_4)$ be any ciphertext of length 4, $m = (x_1, x_2, x_3, x_4)$ the corresponding plaintext and key ...

Example 2. Demonstrate that a one-time pad is not perfectly secure if a key is reused.

A cryptosystem is perfectly secret if

$$\Pr[\mathcal{E}_k(m_0) = c] = \Pr[\mathcal{E}_k(m_1) = c]$$

for all $m_0, m_1 \in \mathcal{M}$ and $c \in \mathcal{C}$. For a one-time pad

$$\begin{aligned} & \Pr[m_0 \oplus k = c] = \Pr[m_1 \oplus k = c] \\ \text{i.e.,} \quad & \Pr[k = c \oplus m_0] = \Pr[k = c \oplus m_1]. \end{aligned}$$

Since $m_0, m_1 \in \mathcal{M}$ and $c \in \mathcal{C}$ vary freely over the respective spaces, the above equation means that k is uniformly distributed over the key space $\mathcal{K}$. Since one key is used twice, this is not the case. Therefore, a one-time pad is not perfectly secure if a key is reused.

Example 3. Show that the Vigenère cipher is perfectly secure if the key is randomly chosen, it is only used once and the plaintext has the same length as the key.

Under the constraints this cipher reduces to a one-time pad without reuse of any secret key. Therefore, this cipher is perfectly secure.

Example 4. Find the reasons for Kirckhoff's principle and discuss the possible counter-arguments.

Kerckhoffs' principle is one of the most influential design principles in cryptography. It states that

> *A cryptographic system should remain secure even if everything about the system, except the secret key, is public knowledge.*

Originally articulated in the 19th century for military ciphers, this principle has become a cornerstone of modern cryptographic design, security proofs, and standardization processes. **Modern Interpretation** Kerckhoffs' principle is best understood as a *design axiom*, not a physical law. It asserts that:

> *All security assumptions must be explicit, minimal, and publicly analyzable.*

In this sense, the principle underlies:

- open cryptographic competitions,
- standardisation processes,
- reproducible cryptanalysis.

While obscurity and secrecy beyond keys may offer tactical advantages, relying on them for core security contradicts both theory and historical evidence. Modern cryptography treats Kerckhoffs' principle not as dogma, but as a disciplined boundary between sound security assumptions and wishful thinking.

Example 5. Let $\mathcal{M}$ be the plaintext space and $\mathcal{K}$ the key space of a perfectly secure encryption scheme. It is reasonable to assume that $\Pr[c] > 0$ for all $c \in \mathcal{C}$. Since the scheme is perfectly secret $\Pr[\mathcal{E}_k(m_0) = c] = \Pr[\mathcal{E}_k(m_1) = c] = \Pr[c \mid m] = \Pr[c]$ for all $m_0 \neq m_1$. Therefore,

$$\Pr[\mathcal{E}_k(m) = c] = \Pr[c]$$

for each $m \in \mathcal{M}$. Therefore, for each pair (m, c) there must exist at least one key $k \in \mathcal{K}$ such that $\mathcal{E}_k(m) = c$ in order to achieve the condition of perfect secrecy. This proves that $|\mathcal{K}| \leq |\mathcal{C}| \leq |\mathcal{M}|$

Example 6. Is a bit permutation of block length n perfectly secure if it is used only once to encrypt a string of length n?

Suppose a bit permutation of block length n is used only once to encrypt a string of length n. The bits of an n-bit block can be numbered from $1, \ldots, n$. A bit permutation $\pi : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ is a one-one onto mapping from $\{1, 2, \ldots, n\}$ onto $\{1, 2, \ldots, n\}$. The set of all such permutations be denoted by S_n which is the de facto key space $\mathcal{K}$. Since the block length is n, the plaintext space is $\mathcal{M} = \{0, 1\}^n$ and the ciphertext space is $\mathcal{C} = \{0, 1\}^n$. For any $m \in \mathcal{M}$ and $c \in \mathcal{C}$, there exists a unique $\pi \in \mathcal{K} = S_n$ such that $\pi(m) = c$. Therefore, for any $m \in \mathcal{M}$ and $c \in \mathcal{C}$

$$\Pr[\mathcal{E}_k(m) = c] = \Pr[\pi(m) = c] = \Pr[\pi] = 2^{-n}.$$

This proves that the scheme is perfectly secret.

3 Computational Security

Assume that the best-known attack against a scheme is exhaustive key search (brute force) and that the key has length n. If testing a single key takes a c CPU cycles and in total

Definition 2. *A scheme is (t, ϵ)-secure if any adversary running in time t (measured in CPU cycles) can break the scheme with probability of ϵ at most.*

Example 7. Assume that the best-known attack against a scheme is exhaustive key search (brute force) and that the key has length n. If testing a single key takes c CPU cycles and in total N CPU cycles are executed, then $\frac{N}{c}$ keys can be tested and the probability of success is approximately $\frac{N}{c2^n}$, if $\frac{N}{c} \ll 2^n$. Hence the scheme is $(N, \frac{N}{c2^n})$ secure.

Definition 3. *An encryption scheme is called computationally secure if every probabilistic algorithm with polynomial running time can only break the scheme with negligible probability in the security parameter n.*

Computational security only provides an *asymptotic security* guarantee, i.e., if the security parameter is sufficiently large.

Example 8. If the best possible attack is a brute-force search of a key of length n and the running time is bounded by $N = p(n)$, where p is a polynomial, then the scheme is $(p(n), \frac{p(n)}{c \cdot 2^n})$-secure, where c is a constant. The probability $\frac{p(n)}{c \cdot 2^n}$ decreases exponentially to zero as n goes to infinity and is thus *negligible*. Therefore, the scheme is computationally secure.

4 Indistinguishable Encryption

We saw that perfect secrecy requires perfect indistinguishability. For a more practical definition we need to relax the requirements.

- We consider only efficient adversaries running in polynomial time.
- We allow a very small advantage over random guesses when an adversary tries to distinguish between messages.
- *Indistinguishability* (IND) means that efficient adversaries are unable to find the correct plaintext out of two possibilities if the ciphertext is given.
- The performance of the adversaries is not noticeably better than random guesses.

We want to define security under *different types of attacks*. The following threat models are considered:

1. adversaries are able to *eavesdrop* on ciphertext messages.
2. adversaries who have access to plaintext/ciphertext pairs (Known Plaintext Attack)
3. adversaries who can choose plaintexts (Chosen Plaintext Attack, CPA)
4. adversaries who can choose ciphertexts and obtain the corresponding plaintext (CCA)

We consider *experiments* (or *games*) between two algorithms, a polynomial-time adversory and a challenger. We denote the adversory by A and the challenger by C. The challenger takes as input a security parameter and sets up the experiment, for example by generating parameters and the keys. C runs the experiments and interacts with A. A has certain choices and capabilities. Finally,

A has to answer a challenge and outputs a single bit. The challenger verifies the answer and outputs 1 (A was successful and won the game) or 0 (A failed). Obviously, A has a 50% chance of randomly guessing the correct answer.

In many security experiments, C chooses a uniform random secret b and A obtains a challenge that depends on b. Finally, A outputs a bit b' and wins the game if $b = b'$. Since the experiment is repeated many times, both b and b' can be considered as random variables. The following table contains all the four combinations of b and b' and their joint probabilities.

	$b' = 0$	$b' = 1$
$b = 0$	$\Pr[b' = 0 \wedge b = 0]$	$\Pr[b' = 1 \wedge b = 0]$
$b = 1$	$\Pr[b' = 0 \wedge b = 1]$	$\Pr[b' = 1 \wedge b = 1]$

Table 1. Joint probabilities of b and b'

We define A's advantage over random guesses as the difference between the probability of success (output of the experiment is 1) and the probability of failure (output of the experiment 0):

$$\begin{aligned}\mathrm{Adv}(A) &= |\Pr[b' = 0 \wedge b = 0] + \Pr[b' = 1 \wedge b = 1] \\ &\quad - \Pr[b' = 1 \wedge b = 0] - \Pr[b' = 0 \wedge b = 1]| \\ &= |\Pr[b' = b] = \Pr[b' \neq b]| \\ &= |\Pr[\mathrm{Out}(C) = 1] - \Pr[\mathrm{Out}(C) = 0]|.\end{aligned}$$

The difference could be negative, so we take the absolute value. A negative advantage would anyway imply a positive advantage for an inverse adversary A' who outputs 1 if A outputs 0 and vice versa. The following are the alternative definitions of the advantage of A:

$$\mathrm{Adv}(A) = 2 \cdot \left|\Pr[b = b'] - \frac{1}{2}\right| = 2 \cdot \left|\Pr[\mathrm{Out}(C) = 1] - \frac{1}{2}\right|,$$
$$\mathrm{Adv}(A) = \big|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]\big|.$$

$$\begin{aligned}\mathrm{Adv}(A) &= |\Pr[b' = b] - \Pr[b' \neq b]| \\ &= |2\Pr[b' = b] - 1| \\ &= 2 \cdot |\Pr[b' = b] - \frac{1}{2}|\end{aligned}$$

The adversary A's advantage is

$$\begin{aligned}
\mathrm{Adv}(A) &= |\Pr[b' = 0 \wedge b = 0] + \Pr[b' = 1 \wedge b = 1] \\
&\quad - \Pr[b' = 1 \wedge b = 0] - \Pr[b' = 0 \wedge b = 1]| \\
&= |\Pr[b' = 0 \wedge b = 0] - \Pr[b' = 1 \wedge b = 0] \\
&\quad + \Pr[b' = 1 \wedge b = 1] - \Pr[b' = 0 \wedge b = 1]| \\
&= |(\Pr[b' = 0 \mid b = 0] - \Pr[b' = 1 \mid b = 0]) \Pr[b = 0] \\
&\quad + (\Pr[b' = 1 \mid b = 1] - \Pr[b' = 0 \mid b = 1]) \Pr[b = 1]| \\
&= |(1 - 2\Pr[b' = 1 \mid b = 0]) \Pr[b = 0] \\
&\quad + (2\Pr[b' = 1 \mid b = 1] - 1) \Pr[b = 1]| \\
&= |(1 - 2\Pr[b' = 1 \mid b = 0]) \frac{1}{2} \\
&\quad + (2\Pr[b' = 1 \mid b = 1] - 1) \frac{1}{2}| \\
&= \frac{1}{2} - \Pr[b' = 1 \mid b = 0] + \Pr[b' = 1 \mid b = 1] - \frac{1}{2} \\
&= \Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0].
\end{aligned}$$

It is assumed that the challenger generates $b \overset{\$}{\leftarrow} \{0, 1\}$. Therefore $\Pr[b = 0] = \Pr[b = 1] = \frac{1}{2}$.

5 Eavesdropping Attack

Definition 4. *Suppose a symmetric encryption scheme is given and consider the following indistinguishability experiment. A challenger takes the security parameter 1^λ as input, generates a key $k \in \mathcal{K}$ by running $Gen(1^\lambda)$ and chooses a random bit $b \overset{\$}{\leftarrow} \{0, 1\}$. A probabilistic polynomial-time adversary A is given 1^λ, but neither k nor b are known to A. The adversary chooses two plaintexts m_0 and m_1 that are equal in length. The challenger returns the ciphertext $\mathcal{E}_k(m_b)$ of one of them. A tries to guess b (i.e., tries of find which of the two plaintexts is encrypted) and outputs b'. The challenger outputs 1 if $b = b'$, and 0 otherwise. The EAV advantage of A is defined as*

$$\mathrm{Adv}^{\mathrm{eav}}(A) = \left|\Pr[b' = b] - \Pr[b' \neq b]\right|.$$

The probability is taken over all random variables in this experiment, i.e., the key k, bit b, encryption $\mathcal{E}_k$ and randomness A.

A scheme is insecure under an EAV attack if the advantage of a smart adversary is not negligible, so that an adversary can successfully derive some information about the plaintext from the ciphertext.

Definition 5. *An encryption scheme has indistinguishable encryption in the presence of an eavesdropper (IND-EAV secure or EAV-secure) if for every probabilistic polynomial-time adversary A, the advantage $\mathrm{Adv}^{\mathrm{eav}}(A)$ is negligible in the security parameter n*

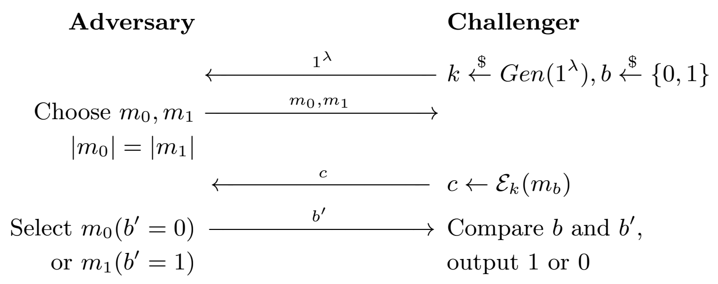

Fig. 1. Indistinguishability experiment in the presence of an eavesdropper

Definition 6. *An encryption scheme is (t, ϵ)-secure in the presence of an eavesdropper if for every probabilistic adversary A running in time t, the advantage of A is less than ϵ:*

$$\mathrm{Adv}^{\mathrm{eav}}(A) < \epsilon.$$

6 Chosen Plaintext Attacks

Definition 7. *Suppose a symmtric encryption scheme is given and an adversary A has access to an encryption oracle. Consider the following experiment. A challenger takes the security parameter 1^λ as input, generates a key $k \in \mathcal{K}$ and chooses a uniform random bit $b \xleftarrow{\$} \{0,1\}$. A probabilistic polynomial-time adversary A is given 1^λ, but k and b are unknown to A. The adversary can choose arbitrary plaintexts and get the corresponding ciphertext from an encryption oracle. The adversary then chooses two different plaintexts m_0 and m_1 of the same length. The challenger returns the ciphertext $\mathcal{E}_k(m_b)$ of one of them. The adversary continues to have the access to the encryption oracle. Finally, A tries to guess b and outputs a bit b'. The challenger outputs 1 if $b = b'$, and 0 otherwise, The IND-CPA advantage of A is defined as*

$$\mathrm{Adv}^{\text{ind-cpa}}(A) = \left|\Pr[b' = b] - \Pr[b' \neq b]\right|.$$

The probability is taken over all random variables in this experiment, i.e., the key k, bit b, encryption $\mathcal{E}_k$ and randomness of A.

7 Chosen Ciphertext Attacks

Suppose a symmetric encryption scheme is given. Consider the following experiment. On input 1^n a challenger generates a random key $k \in \mathcal{K}$ and a random bit $b \xleftarrow{\$} \{0,1\}$. A probabilistic polynomial-time adversary is given 1^n, but k is not known to A. The adversary can ask an oracle to encrypt arbitrary plaintexts

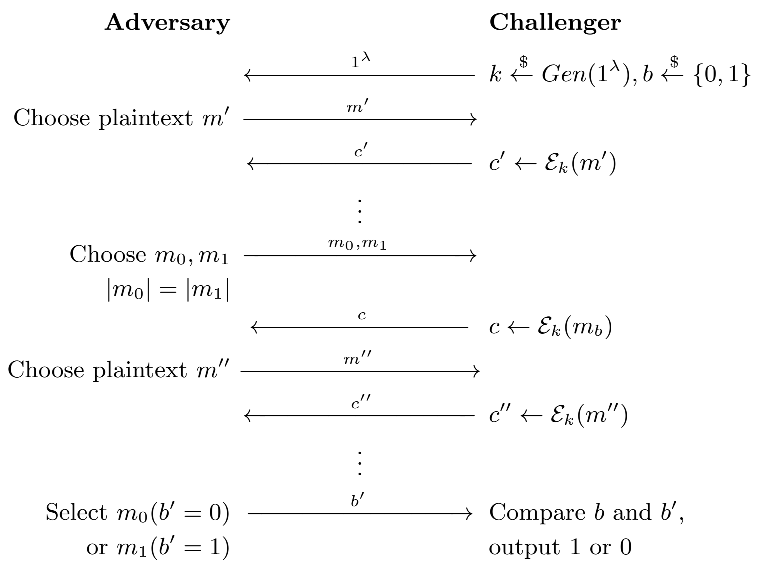

Fig. 2. CPA Indistinguishability experiment. The adversary may repeatedly ask for the encryption of chosen plaintexts m', m''.

and to decrypt ciphertexts. The adversary chooses two different plaintexts m_0 and m_1 of the same length. The challenger returns the ciphertext $c = \mathcal{E}_k(m_b)$ of one of them. The adversary A continues to have access to the encryption and decryption oracle, only decryption of the challenger text c is not permitted. Finally, A tries to guess b and outputs a bit b'. The challenger outputs 1 if $b = b'$, and 0 otherwise. Then the IND-CCA advantage of A is defined by

$$\mathrm{Adv}^{\text{ind-cca}}(A) = \left|\Pr[b' = b] - \Pr[b' \neq b]\right| .$$

The probability is taken over all random variables in this experiment, i.e., the key k, bit b, encryption $\mathcal{E}_k$ and randomness of A.

Definition 8. *A scheme has indistinguishable encryptions under adaptive chosen ciphertext attack (IND-CCA2 secure or CCA2-secure) if for every probabilistic polynomial-time adversary A, the advantage* $\mathrm{Adv}^{\text{ind-cca}}(A)$ *is negligible in n.*

In the IND-CCA1 experiment, the attack is not adaptive, and the adversary may use the decryption oracle only before being given the challenge. In contrast, the IND-CCA2 experiment allows the adversary to adapt their queries to the challenge. CCA2 security is stronger than CCA1 security, and CCA security mostly refer to the CCA2 experiment.

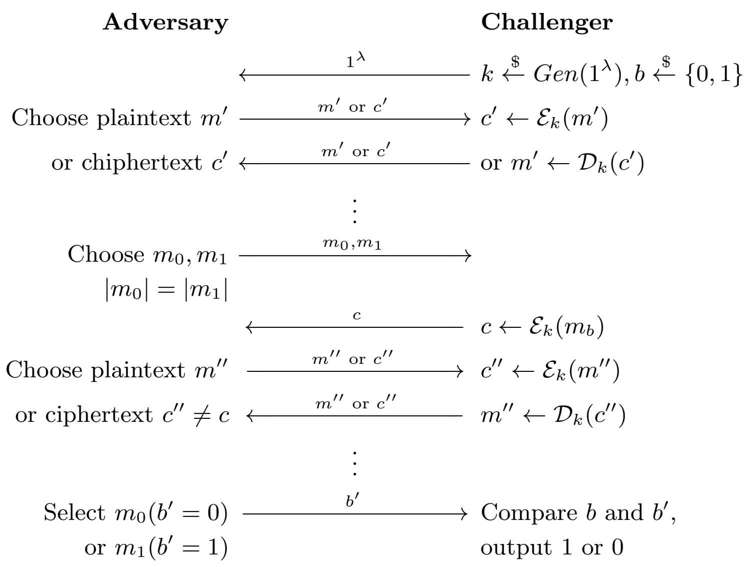

Fig. 3. CCA2 indistinguishability experiment. The adversary may repeatedly ask for the encryption of chosen plaintexts m', m'' and for the decryption of the chosen ciphertexts c', c'' except the challenge c.

8 Pseudorandom Generators

Definition 9. *Let G be a deterministic polynomial-time algorithm that takes an input seed $s \in \{0,1\}^n$ and outputs a string $G(s) \in \{0,1\}^{l(n)}$, where $l(\cdot)$ is a polynomial and $l(n) > n$ for all $n \in \mathbb{N}$. The G is called a pseudorandom generator (prg) is no output can be distinguished from a uniform random sequence in polynomial time.*

Consider the following experiment: on input 1^n a see $s \xleftarrow{\$} \{0,1\}^n$ and a bit $b \xleftarrow{\$} \{0,1\}$ are chosen uniformly at random. A probabilistically polynomial-time adversary A obtains 1^n, but knows neither s not b. A challenger chooses a uniform random string $r \xleftarrow{\$} \{0,1\}^{l(n)}$. If $b = 0$ then $c = r$ is given to A. Otherwise, A receives $c = G(s)$. The adversary tries to distinguish between the two cases and outputs a bit b'. The challenger outputs 1 if $b = b'$, and 0 otherwise. Then the prg-advantage of A is defined as

$$\mathrm{Adv}^{\mathrm{prg}}(A) = \left|\Pr[b' = b] - \Pr[b \neq b']\right| .$$

The probability is taken over all random variables in this experiment, i.e., the seed s, bit b, string r and the randomness of A.

G is called a pseudorandom generator if for all probabilistic polynomial-time distinguisher A, the prg-avangage is negligible in n.

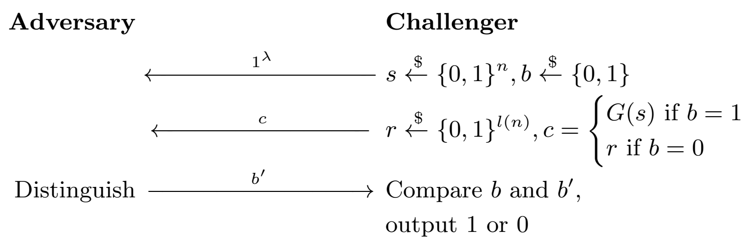
Adversary
Challenger
1^λ
$s \overset{\$}{\leftarrow} \{0,1\}^n, b \overset{\$}{\leftarrow} \{0,1\}$
c
$r \overset{\$}{\leftarrow} \{0,1\}^{l(n)}, c = \begin{cases} G(s) \text{ if } b = 1 \\ r \text{ if } b = 0 \end{cases}$
Distinguish
b'
Compare b and b', output 1 or 0

Fig. 4. Distinguishability experiment for a pseudorandom generator G.

Theorem 2. *A generator is peusodorandom if and only if it is unpredictable in polynomial time.*

Proof. Insert the proof □

The output of a pseudorandom generator can be used as a keystream. Like the one-time pad, xor-ing the plaintext and the keystream defines a fixed-length encryption scheme. This type of scheme is called a stream cipher.

Definition 10. *Let $l(\cdot)$ be a polynomial. A pseudorandom generator G, which on input $k \in \{0,1\}^n$ produces an output sequence $G(k) \in \{0,1\}^{l(n)}$, defines a fixed-length encryption scheme by the following construction:*

The key generation algorithm $Gen(1^n)$ takes the security parameter 1^n as input and outputs a uniform random key $k \overset{\$}{\leftarrow} \{0,1\}^n$ of length n. Set $\mathcal{M} = \mathcal{C} = \{0,1\}^{l(n)}$. Encryption $\mathcal{E}_k$ and decryption $\mathcal{D}_k$ are identical and are defined by xor*-ing the output stream $G(s)$ with the input data:*

$$c = \mathcal{E}_k(m) = m \oplus G(k) \text{ and } m = \mathcal{D}_k(c) = c \oplus G(k).$$

Theorem 3. *If G is a pseudorandom generator, then the associated encryption scheme has indistinguishable encryptions in the presence of an eavesdropper (EAV-secure).*

Proof. Suppose the encryption scheme is not EAV-secure; then there is a polynomial-time algorithm A with a non-negligible advantage $\mathrm{Adv}^{\mathrm{eav}}(A)$ in the EAV experiment. We construct an adversary B in the prg distinguishability experiment which uses A as a subroutine.

□

References

1. Hoffstein, J., Pipher, J., Silverman, J.H.: An Introduction to Mathematical Cryptography. Undergraduate Texts in Mathematics, Springer, New York, NY, USA (2008)

2. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. CRC Press, Boca Raton, FL, 3rd edn. (2020)
3. Knospe, H.: A Course in Cryptography, Pure and Applied Undergraduate Texts, vol. 40. American Mathematical Society, Providence, RI, USA (2019), https://bookstore.ams.org/amstext-40
4. Stinson, D.R., Paterson, M.B.: Cryptography: Theory and Practice. Textbooks in Mathematics, Chapman & Hall/CRC, Boca Raton, FL, USA, 4 edn. (2017)