Lecture 1 – From Perfect Security to Blockciphers

François Dupressoir

2023

We first consider a simple setting: Aniket and Barbara want to securely exchange a single message whose length they know in advance. This setting gives rise to simple constructions—*enciphering schemes* that we use to introduce a number of basic and standard methods in modern cryptography.

First, we'll specify which algorithms can be considered enciphering schemes by defining their *syntax*. Then, we'll specify the most basic properties such enciphering schemes should possess: *correctness*. Finally, and most importantly, we will discuss what exactly it might mean for an enciphering scheme to be *secure*.

This will lead us to the modern practice of game-based (or property-based) definitions of security.

1.1 Enciphering Schemes: Syntax and Correctness

Informally, we are interested in taking a message in plain text—often referred to as the *plaintext*, taken from some *message space* $\mathcal{M}$—and some key—taken from some *key space* $\mathcal{K}$; and outputs an enciphered message—often referred to as *the ciphertext*—taken from some *cipher space* $\mathcal{C}$; in such a way that the original message can be recovered given knowledge of ciphertext and key, but not without the key.

An enciphering scheme (for us) is such that $\mathcal{M} = \mathcal{C}$, and is specified by three algorithms:

- A probabilistic algorithm that generates keys in $\mathcal{K}$;
- An algorithm that *enciphers* a plaintext under a key, and into a ciphertext; and
- An algorithm that *deciphers* a ciphertext under a key, and into a plaintext.

This exactly specifies the syntax of enciphering scheme. The formal definition (Definition 1.1) does a bit more legwork in introducing some notation, and giving names to those algorithms. This will later give us nice ways of abstracting over specific enciphering schemes.

Definition 1.1 (Symmetric Enciphering Scheme)**.** A *symmetric enciphering scheme* E is a triple of algorithms Kg, E, and D, where:

- Kg randomly generates a $k \in \mathcal{K}$;

- E takes a key k and a message $m \in \mathcal{M}$ to output ciphertext $c \leftarrow \mathsf{E}_k(m) \in \mathcal{C}$, with $\mathcal{C} = \mathcal{M}$; and
- D takes a key k and a ciphertext $c \in \mathcal{C}$ and to output a purported message $m' \leftarrow \mathsf{D}_k(c)$.

With this definition in place, we can generically define what it means for an enciphering scheme to be *correct*.

Definition 1.2 (Correctness of Enciphering Schemes)**.** An enciphering scheme $E = (\mathsf{Kg}, \mathsf{E}, \mathsf{D})$ is correct iff, for all $k \in \mathcal{K}$ and $m \in \mathcal{M}$, we have $\mathsf{D}_k(\mathsf{E}_k(m)) = m$.

1.2 The One-Time Pad

The one-time pad is a very simple enciphering scheme where keys, messages and ciphertexts are all bitstrings of some fixed length ℓ. Figure 1.1 specifies its algorithms Kg, E and D for some $\ell > 0$. Note also the notation—which will become pervasive—for sampling a value x uniformly at random in a (finite) set S: $x \leftarrow_{\$} S$. (We will also use it to denote storing in a variable the result of running a probabilistic algorithm.) $\oplus$ is bitwise exclusive or (XOR).

$\mathsf{Kg}()$	$\mathsf{E}_k(m)$	$\mathsf{D}_k(c)$
$k \leftarrow_{\$} \{0,1\}^{\ell}$	$c \leftarrow m \oplus k$	$m \leftarrow c \oplus k$
return k	**return** c	**return** m

Figure 1.1: The one-time pad; ℓ is the intended message length

1.3 Security of Enciphering Schemes

A natural question, then is: how much *security* does such a simple construction as the one-time pad provide? The answer, as we see next, is simultaneously "it provides perfect security," and "it provides no security whatsoever". The main crux of what looks like a paradox right now, is that we do not even know what it means for an enciphering scheme to be secure. Let's remedy that.

1.3.1 Key Recovery

By Kerckhoffs' principle, we must certainly ensure that the key cannot be recovered from the system—if an adversary can recover the key from a ciphertext—and is assumed to know all details of the algorithm used, then they can surely decipher that ciphertext as well.

Adversary Goal So we first attempt to define what it means for an enciphering scheme (*any* enciphering scheme) to be secure against key recovery. We do so using a *security experiment* (or *security game*), and defining an *adversarial advantage*—which essentially measures the *insecurity* of a scheme against an adversary.

Figure 1.2: The passive key-recovery game $\mathsf{Exp}_E^{\text{kr-pas}}()$, and the "guessing" adversary $\mathbb{A}_{\text{guess}}$ (right).

Definition 1.3 (Passive Key Recovery Security for Enciphering Schemes)**.** Let E be an enciphering scheme. The *advantage of* $\mathbb{A}$ *in passively recovering the key* is defined as follows, for the experiment $\mathsf{Exp}_E^{\text{kr-pas}}()$ defined in Figure 1.2.

$$\mathsf{Adv}_E^{\text{kr-pas}}(\mathbb{A}) \stackrel{def}{=} \Pr\left[\mathsf{Exp}_E^{\text{kr-pas}}(\mathbb{A}) : \widehat{k} = k^*\right]$$

An enciphering scheme E is said to be (t, ϵ)*-secure against passive key recovery* if, for every algorithm $\mathbb{A}$ running in time at most t, we have $\mathsf{Adv}_E^{\text{kr-pas}}(\mathbb{A}) \leq \epsilon$.

With this definition of security, it is clear that the adversary cannot do much: they get to see nothing that depends on the key, so the best they can do is guess. Such an adversary is given on the right hand side of Figure 1.2: they simply run the key generation algorithm, and hope it outputs the same key. If, say, Kg samples its output uniformly at random in the key space $\mathcal{K}$, then $\mathsf{Adv}_E^{\text{kr-pas}}(\mathbb{A}_{\text{guess}}) = 1/|\mathcal{K}|$.

Adversarial powers Clearly, beyond allowing us to introduce concepts and notations slowly, passive key recovery isn't very interesting as a security notion. Can the adversary recover the key when observing a ciphertext? We certainly hope not! But what other powers could we give the adversary?

Figure 1.3 shows three different experiments for key recovery under increasing adversary powers: (one-time) *known ciphertext attacks* (kr-1kca), (one-time) *known plaintext attack* (kr-1kpa), and (one-time) *chosen plaintext attack*. The shape of their advantage expression is determined entirely by the goal of key recovery: the only thing that changes is which game the adversary plays, but their winning condition is the same.

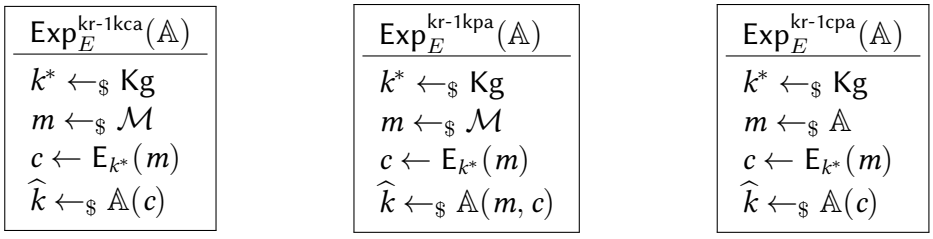

Figure 1.3: Adding one-time powers to the key-recovery game in three different ways.

1.3.2 One-Wayness

The goal of enciphering is not to protect the key, but to protect the plaintext. How can we express that no adversary can reasonably do so? Let us first consider the notion of one-wayness, which captures the idea that recovering the plaintext *in full* from the ciphertext should be hard.

$\mathsf{Exp}_E^{\text{ow-pas}}(\mathbb{A})$

$k \leftarrow_\$ \mathsf{Kg}$
$m^* \leftarrow_\$ \mathcal{M}$
$c^* \leftarrow \mathsf{E}_k(m^*)$
$\widehat{m} \leftarrow_\$ \mathbb{A}(c^*)$

Figure 1.4: Passive one-time one-way security for symmetric enciphering scheme E

Definition 1.4 (Passive One-Time One-Way Security for Enciphering Schemes)**.** Let E be an enciphering scheme. We define the *advantage of* $\mathbb{A}$ *in passively breaking one-wayness* as follows, for the experiment $\mathsf{Exp}_E^{\text{ow-pas}}()$ defined in Figure 1.4.

$$\mathsf{Adv}_E^{\text{ow-pas}}(\mathbb{A}) = \Pr\left[\mathsf{Exp}_E^{\text{ow-pas}}(\mathbb{A}) : \widehat{m} = m^*\right]$$

An enciphering scheme E is said to be (t, ϵ)*-secure against passive one-wayness attacks* if, for every algorithm $\mathbb{A}$ running in time at most t, we have $\mathsf{Adv}_E^{\text{ow-pas}}(\mathbb{A}) \leq \epsilon$.

1.3.3 Perfect Secrecy

Ensuring that no adversary is able to recover the message in full is nice. Ensuring that the adversary learns *no information* about the message at all is nicer.

Definition 1.5 captures this by expressing the fact that, whatever distribution the message is sampled from, the distribution over ciphertexts induced by enciphering a random plaintext under a freshly generated key is independent from the plaintext being enciphered.

Definition 1.5 (Perfect Secrecy)**.** A symmetric enciphering scheme E satisfies perfect secrecy if and only if for all message distributions over $\mathcal{M}$, the following holds.

$$\forall c \in \mathcal{C}, m \in \mathcal{M}. \Pr\left[m^* = m \mid c^* = c\right] = \Pr\left[m^* = m\right]$$

The probabilities are taken over $k \leftarrow_\$ \mathsf{Kg}$ and $m^* \leftarrow_\$ \mathcal{M}$ (according to the aforementioned message distribution), which fixes $c^* = \mathsf{E}_k(m^*)$.

Security of the One-Time Pad The One-Time Pad turns out to be perfectly secure.

Theorem 1.1 (Security of the One-Time Pad)**.** *The One-Time Pad satisfies perfect security.*

Shannon's Theorem Unfortunately for us, the One-Time Pad turns out to be (up to isomorphism) the only enciphering scheme that is perfectly secure.

Theorem 1.2 (Shannon's Theorem)**.** *Let $E = (\mathsf{Kg}, \mathsf{E}, \mathsf{D})$ be an enciphering scheme with $\mathcal{K} = \mathcal{M}$. Then E is perfectly secure iff* Kg *draws from $\mathcal{K}$ uniformly at random and E satisfies that for all (m, c) pairs, there exists a unique key k such that $\mathsf{E}_k(m) = c$.*

1.3.4 Indistinguishability

We want to weaken perfect secrecy a little bit so that more schemes satisfy the notion, but without weakening it so much as to make it meaningless. We've already seen a few notions that allowed a bit of sloppiness. Can we express perfect secrecy as a game, then loosen it a little bit?

There are a few equivalent ways of expressing perfect secrecy. That given in Definition 1.5 is the original one given by Shannon [Sha49]. However, it is not directly useful to express security as a game: it quantifies over the plaintext distribution, and it directly talks about the independence of some distribution.

Definition 1.6

Definition 1.6 (Perfect Indistinguishability)**.** A symmetric enciphering scheme E satisfies perfect indistinguishability if and only if the following holds.

$$\forall c \in \mathcal{C}, m \in \mathcal{M}. \Pr\left[c^* = c \mid m^* = m\right] = |\mathcal{C}|^{-1}$$

The probability is taken over $k \leftarrow_\$ \mathsf{Kg}$.

Theorem 1.3. *An enciphering scheme has perfect secrecy if and only if it has perfect indistinguishability.*

We can express this property as a game—however, the adversary's goal here is no longer to *recover* or compute a value, but to distinguish two different experiments.

$\mathsf{Exp}_E^{\text{1ind-real}}(\mathbb{A})$
$k \leftarrow_\$ \mathsf{Kg}$
$m \leftarrow_\$ \mathbb{A}$
$c^* \leftarrow \mathsf{E}_k(m)$
$\widehat{b} \leftarrow_\$ \mathbb{A}(c^*)$

$\mathsf{Exp}_E^{\text{1ind-ideal}}(\mathbb{A})$
$k \leftarrow_\$ \mathsf{Kg}$
$m \leftarrow_\$ \mathbb{A}$
$c^* \leftarrow_\$ \mathcal{C}$
$\widehat{b} \leftarrow_\$ \mathbb{A}(c^*)$

Figure 1.5: One-time indistinguishability

Definition 1.7 (Game-Based Perfect Indistinguishability)**.** Let E be an enciphering scheme. We define the *advantage of $\mathbb{A}$ in one-time distinguishing E from random* as follows, for the experiments $\mathsf{Exp}_E^{\text{1ind-real}}()$ and $\mathsf{Exp}_E^{\text{1ind-ideal}}()$ defined in Figure 1.5.

$$\mathsf{Adv}_E^{\text{1ind}}(\mathbb{A}) = \Pr\left[\mathsf{Exp}_E^{\text{1ind-real}}(\mathbb{A}) : \widehat{b} = 1\right] - \Pr\left[\mathsf{Exp}_E^{\text{1ind-ideal}}(\mathbb{A}) : \widehat{b} = 1\right]$$

An enciphering scheme E is said to be *perfectly indistinguishable* if, for every algorithm $\mathbb{A}$, we have $\mathsf{Adv}^{\mathsf{iind}}_{E}(\mathbb{A}) = 0$.

Now *this* definition can effectively be weakened in two ways: first, we can—instead of considering all possible algorithms—only consider adversaries with bounded resources, as we did for key recovery and one-wayness; and second, we can—instead of requiring that the adversary's advantage be 0—consider a scheme secure if any (bounded) adversary's advantage is small.

Definition 1.8 (Game-Based Indistinguishability)**.** An enciphering scheme E is said to be (t, ϵ)-*indistinguishable* if, for every algorithm $\mathbb{A}$ that runs in time at most t, we have $\mathsf{Adv}^{\mathsf{iind}}_{E}(\mathbb{A}) \leq \epsilon$.

This will be our baseline security notion for confidentiality in the rest of this unit, and serves as the (heuristic) assumption the entirety of practical cryptography relies on: that *blockciphers* are indistinguishable from random permutations.

1.4 Blockciphers

To do anything worth while, we need to allow ourselves to define security when the same key can be used multiple times—recall that we've so far only defined notions under one-time attacks! We take the smallest possible step in this direction by defining blockciphers.

Definition 1.9 (Blockcipher)**.** A *blockcipher* E with *block length* ℓ is a symmetric enciphering scheme with $\mathcal{M} = \mathcal{C} = \{0,1\}^{\ell}$.

Lemma 1.4 (Blockciphers as Keyed Permutations)**.** *Let* $E = (\mathsf{Kg}, \mathsf{E}, \mathsf{D})$ *be a* correct *blockcipher. Then, for any fixed key* k *output by* Kg*, the enciphering function* E_k *is a permutation.*

Proof. Left as an exercise to the reader, recalling the definition of correctness for enciphering schemes. □

We do not discuss the construction (or *realisation*) of blockciphers in this unit. The second (optional) half of the worksheet explores this question in depth—mostly negatively, to show that designing a good blockcipher is hard and that you are strongly encouraged to show humility if you try. But let us consider instead what kind of security we might want, how we can define it, and the boringest ways in which we can break it—this will inform some design constraints.

1.4.1 Security: Key Recovery

Let us first revisit key recovery under chosen plaintext attacks. Unlike previously—where we were considering one-time security notions, we now want to allow the adversary to interact with the key *multiple times*—but we still can't give the adversary they key!

The solution here is to consider adversaries that have *oracle access* to the encryption algorithm with a fixed key: this controls the way in which the adversary is allowed to make

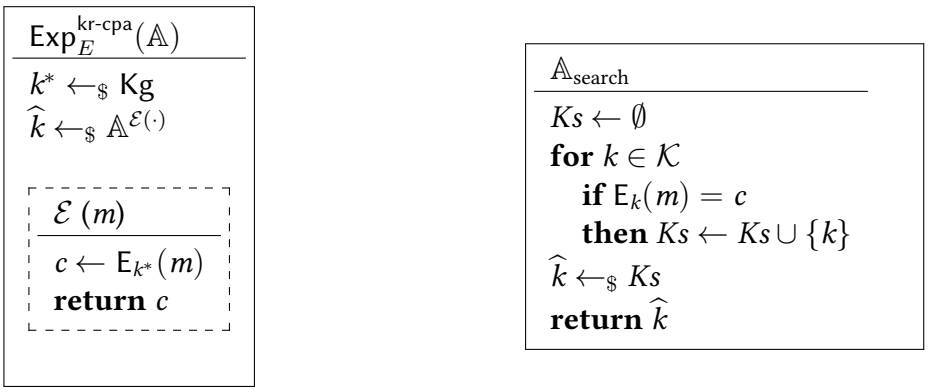

Figure 1.6: The "key-recovery under chosen plaintext attack game" (left) and the "exhaustive search" adversary that uses a single known-plaintext-ciphertext pair (right).

use of the key in a minimally intrusive way. In particular, unless a specific note is made otherwise, the adversary can choose their queries to their oracles *adaptively*: make a query, see the result, *then* choose the next query.

Definition 1.10 (Key Recovery Security for Blockciphers)**.** Let E be a blockcipher. We define the *advantage of* $\mathbb{A}$ *in recovering the key from* E *under chosen plaintext attack* as follows, where experiment $\mathsf{Exp}^{\text{kr-cpa}}_E(\mathbb{A})$ is defined in Figure 1.6.

$$\mathsf{Adv}^{\text{kr-cpa}}_E(\mathbb{A}) = \Pr\left[\mathsf{Exp}^{\text{kr-cpa}}_E(\mathbb{A}) : \widehat{k} = k^*\right]$$

E is said to be (t, q, ϵ)*-secure against chosen plaintext key recovery* if, for every algorithm $\mathbb{A}$ running in time at most t and making at most q queries to its chosen plaintext oracle $\mathcal{E}(\cdot)$, we have $\mathsf{Adv}^{\text{kr-cpa}}_E(\mathbb{A}) \leq \epsilon$.

Exhaustive search as baseline security level. A simple (but costly) attack, given one or several plaintext-ciphertext pairs, is to simply iterate through all the keys and eliminate those that fail to map the plaintexts to the corresponding ciphertexts.

The number of encipherings it takes to run an exhaustive search (or rather, its base 2) is often used as a baseline for the security of a blockcipher. When a blockcipher's actual key recovery security strays a bit too far from this then the blockcipher is considered broken. This gives somewhat uniform measures for all notions of security: "we want 256-bit security" translates to "we want breaking whatever security we just asked you to obtain to be as costly as running 2^{256} encipherings of something". However, it's a lot less uniform in practice than we'd like—as a science—to pretend.

Current recommendations: if you really can't do anything else, 112 bit security is OK (lightweight cryptography); if you don't really care about the data long-term but need to show you did something short-term, 128 bit security is what you want; if you care about the data long-term, you must aim for 256 bit security.

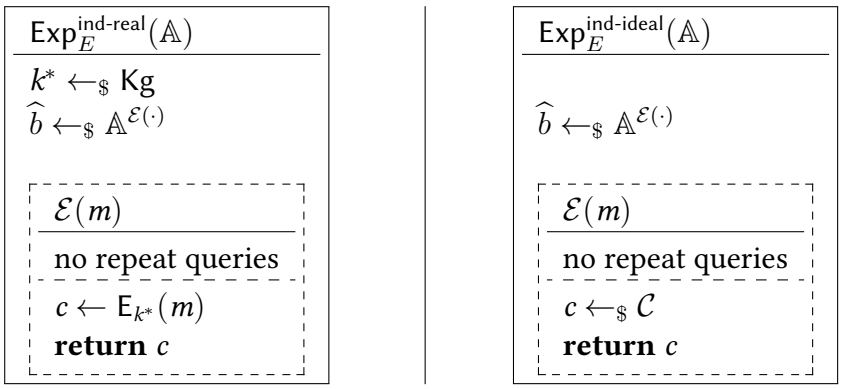

Figure 1.7: The real and ideal indistinguishability experiments.

1.4.2 Pseudorandomness

As before, key recovery is a nice baseline, but what we want is *indistinguishability*. For blockciphers, for some reason, it's called pseudorandomness.

Definition 1.11 (Pseudorandom Permutation)**.** Let E be a blockcipher. We define the *advantage of* $\mathbb{A}$ *in distinguishing* E *from a random permutation* as follows, where experiments $\mathsf{Exp}^{\text{ind-real}}_E(\mathbb{A})$ and $\mathsf{Exp}^{\text{ind-ideal}}_E(\mathbb{A})$ are defined in Figure 1.7.

$$\mathsf{Adv}^{\text{ind}}_E(\mathbb{A}) = \Pr\left[\mathsf{Exp}^{\text{ind-real}}_E(\mathbb{A}) : \widehat{b} = 1\right] - \Pr\left[\mathsf{Exp}^{\text{ind-ideal}}_E(\mathbb{A}) : \widehat{b} = 1\right]$$

E is said to be a (t, q, ϵ)*-secure pseudorandom permutation* if, for every algorithm $\mathbb{A}$ running in time at most t and making at most q queries to its $\mathcal{E}(\cdot)$ oracle, we have $\mathsf{Adv}^{\text{ind}}_E(\mathbb{A}) \leq \epsilon$.

Note here that we *must* restrict the adversary from querying the same input twice to the $\mathcal{E}$ oracle: in the real world, they would get the same response to both identical queries; in the ideal world, they would only get the same response with low probability—a clear and trivial distinguishing attack!

The birthday bound. What we cannot rule out immediately is repeat responses: those never happen in the real world, but could happen in the ideal world. This is known as a *collision*—two messages $m \neq m'$ such that $\mathcal{E}(m) = \mathcal{E}(m')$. An adversary that makes q queries to a random permutation will find such a collision with probability roughly $\frac{q \cdot (q-1)}{2 \cdot |\mathcal{C}|}$ (the birthday bound).

If exhaustive search placed a constraint on key size, the birthday bound places a constraint on the block length ℓ of a blockcipher if we are hoping for it to be pseudorandom.

Bibliography

[Sha49] Claude Elwood Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.