Block ciphers
And modes of operation

Foundations of Cryptography
Computer Science Department
Wellesley College

Table of contents

Introduction

Pseudorandom permutations

Block Ciphers

Modes of Operation

CCA-Secure
Keyed permutations: Some definitions

Definition. Let $F : \{0, 1\}^* \times \{0, 1\}^* \to \{0, 1\}^*$ be an efficient, length-preserving, keyed function. We call $F$ a **keyed permutation** if for every $k$, the function $F_k(\cdot)$ is one-to-one.

Definition. We say that a keyed permutation is **efficient** if there is a polynomial time algorithm computing $F_k(x)$ given $k$ and $x$, as well as a polynomial-time algorithms computing $F_k^{-1}(x)$ given $k$ and $x$.

Remark. The input and output lengths, called the **block size** are the same, but the key length may be smaller or larger than the block size.

Pseudorandom permutations

Definition. Let $F : \{0, 1\}^* \times \{0, 1\}^* \to \{0, 1\}^*$ an efficient keyed permutation. We say that $F$ is a **pseudorandom permutation** if for all probabilistic polynomial-time distinguishers $D$, there exists a negligible function $\text{negl}$ such that:

$$\left| \Pr[D_{F_k(\cdot)}(1^n) = 1] - \Pr[D_f(\cdot)(1^n) = 1] \right| \leq \text{negl}(m),$$

where $k \leftarrow \{0, 1\}^n$ is chosen uniformly at random and $f$ is chosen uniformly at random from the set of permutations mapping $n$-bit strings to $n$-bit strings.


**Pseudorandom functions and permutations are polynomially indistinguishable**

*Theorem 3.27.* If $F$ is a pseudorandom permutation then it is also a pseudorandom function.

*Proof.* The basic idea behind the proof is that a random function $f$ looks identical to a random permutation unless a distinct pair of values $x$ and $y$ are found for which $f(x) = f(y)$. The probability of finding such points $x, y$ using a polynomial number of queries is negligible.

---

**A new security concern**

- If $F$ is an efficient pseudorandom permutation then cryptographic schemes based on $F$ might require honest parties to compute both $F_k$ and $F_k^{-1}$.
- We may wish that $F_k$ is indistinguishable from a random permutation even if the distinguisher is given oracle access to the inverse permutation.
**Strong pseudorandom permutations**

*Definition 3.28.* Let \( F : \{0, 1\}^* \times \{0, 1\}^* \to \{0, 1\}^* \) an efficient keyed permutation. We say that \( F \) is a **strong pseudorandom permutation** if for all probabilistic polynomial-time distinguishers \( D \), there exists a negligible function \( \text{negl} \) such that:

\[
\left| \Pr[D_{F_k}^k(\cdot), F_k^{-1}(\cdot)(1^n) = 1] - \Pr[D_{f}^f(\cdot), f^{-1}(\cdot)(1^n) = 1] \right| \leq \text{negl}(m),
\]

where \( k \leftarrow \{0, 1\}^n \) is chosen uniformly at random and \( f \) is chosen uniformly at random from the set of permutations mapping \( n \)-bit strings to \( n \)-bit strings.

---

**Block ciphers**

- From your reading you know that stream ciphers can be modeled as pseudorandom generators.
- The analogue for the case of a strong pseudorandom permutation is a **block cipher**.
- Block ciphers are not secure encryption schemes. Rather, they are building blocks that can be used to construct secure schemes.
A mode of operation is essentially a way of encrypting arbitrary-length messages using a block cipher (i.e., pseudorandom permutation).

Note that messages can be unambiguously padded to a total length that is a multiple of the block size by appending a 1 followed by sufficiently many 0’s. Our notation: $\ell$ blocks each of size $n$.

*Donald has volunteered to help demonstrate.

---

**Electronic code book (ECB) mode**

*Electronic code book.* Given a plaintext message $m = m_1, \ldots, m_\ell$, the ciphertext is obtained by "encrypting" each block separately, i.e., $c = \langle F_k(m_1), \ldots, F_k(m_\ell) \rangle$. 
Donald does ECB

- ECB is deterministic and therefore cannot be CPA-secure.
- Worse, ECB-mode encryption does not even have indistinguishable encryptions in the presence of eavesdroppers.

*Uncompressed bitmap format encrypted using AES in ECB mode.

Cipher block chaining (CBC) mode

Cipher block chaining. We choose a random initial vector (IV) of length $n$. The ciphertext is obtained by applying the pseudorandom permutation to the XOR of the current plaintext block and the previous ciphertext block. That is, we set $c_0 = IV$ and then, for $i = 1$ to $\ell$, set $c_i = F_k(c_{i-1} \oplus m_i)$. 
Introduction

Pseudorandom permutations

Block Ciphers

Modes of Operation

Donald disappears*

- Encryption in CBC is probabilistic and it has been proven that if $F$ is a pseudorandom permutation then CBC-mode encryption is CPA-secure.

- All is not rosie in cipherland however: Encryption must be carried out sequentially. If parallel processing is available CBC may not be the most efficient choice.

*Uncompressed bitmap format encrypted using AES in CBC mode.

Output feedback (OFB) mode

Output Feedback mode. Again a random initial vector ($IV$) of length $n$ is chosen and a stream is generated from $IV$ as follows: Define $r_0 := IV$ and set the $i$th block of the stream $r_i = F_k(r_{i-1})$. Then, for $i = 1$ to $\ell$, set $c_i = m_i \oplus r_i$. 

![Diagram of OFB mode]
Donald stays away

- This mode is also probabilistic and it and can be shown to be CPA-secure.
- Both encryption and decryption must be carried out sequentially, but the bulk of the computation can be carried out independently of the message.

*Namely computing the pseudorandom stream.

Counter (CTR) mode

Randomized counter mode. A random initial vector (IV) of length $n$ is chosen, this is referred to as ctr. Then a stream is generated from IV by computing $r_i := F_k(ctr + i)$ where ctr and $i$ are viewed as binary numbers and addition is performed modulo $2^n$. Finally, the $i$th ciphertext block is computed as $c_i = m_i \oplus r_i$.

*There are a number of different types of counter modes.
Donald gets tired of waiting in the wings

- Again this mode is probabilistic and it and can be shown to be CPA-secure.
- Both encryption and decryption can be fully parallelized and, as with OFB mode, it is possible to generate the pseudorandom stream ahead of time.
- Finally, it is possible to decrypt the $i$th block of ciphertext without decrypting anything else*.

* A property known as random access.

Randomized counter (CTR) mode is CPT-secure

**Theorem 3.29.** If $F$ is a pseudorandom function, then randomized counter mode has indistinguishable encryption under a chosen-plaintext attack.

**Proof.** As previously, we prove counter mode is CPT-secure when a truly random function is used. We then prove that replacing the random function by a pseudorandom function cannot make the scheme insecure.

Let $\text{ctr}^*$ denote the initial value used when the challenge ciphertext is encrypted in the $\text{PrivK}^{\text{cpa}}$ experiment. When a random function is used in CTR mode, security is achieved as long as each block $c_i$ is encrypted using a value $\text{ctr}^* + i$ that was never used by the encryption oracle in to answer any of its queries since in this case $f(\text{ctr}^* + i)$ is completely random.
The tilde experiment

Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ denote the randomized counter mode encryption scheme, and let $\tilde{\Pi} = (\tilde{\text{Gen}}, \tilde{\text{Enc}}, \tilde{\text{Dec}})$ be the encryption scheme that is identical to $\Pi$ except that a truly random permutation $f$ is used in place of $F_k$.

We claim that for every PPT adversary $A$ there exists a negligible function negl such that

$$\Pr\left[\text{PrivK}_{A,\tilde{\Pi}}^\text{cpa}(n) = 1\right] \leq \frac{1}{2} + \text{negl}.$$ 

The adversary is polynomially bounded

Let $q$ be a polynomial-upper bound on the number of oracle queries made by $A$ as well as on the maximum length of any such query. As above let $\text{ctr}^*$ denote the initial value used when the challenge ciphertext is encrypted, and let $\text{ctr}_i$ the initial value used when the $i$th oracle query is answered.

When the challenge ciphertext is encrypted, $f$ is applied to the values

$$\text{ctr}^* + 1, \ldots, \text{ctr}^* + \ell^*$$

while when the $i$th is answered, the function $f$ is applied to the values

$$\text{ctr}_i + 1, \ldots, \text{ctr}_i + \ell_i.$$

In all cases the number of blocks is bounded by $q(n)$. 
There are two cases to consider

1. *Suppose there do not exist any* \( i, j, j' \geq 1 \) *for which* \( \text{ctr}_i + j = \text{ctr}^* + j' \). Then, the values \( f(\text{ctr}^* + 1), \ldots, f(\text{ctr}^* + \ell^*) \) are independently and uniformly distributed since \( f \) was not applied to any of these when encrypting oracle queries. The challenge text is encrypted with a random string and the probability that \( \mathcal{A} \) outputs \( b' = b \) is exactly 1/2 as in the one-time pad.

2. *There exists* \( i, j, j' \geq 1 \) *for which* \( \text{ctr}_i + j = \text{ctr}^* + j' \). In this case \( \mathcal{A} \) has it made in the shade since it can easily determine the value \( f(\text{ctr}_i + j) = f(\text{ctr}^* + j') \) from the answer to its \( i \)th oracle query. We analyze the probability that this occurs.

**Probability of overlaps continued**

The probability is as large as possible when \( \ell^* = \ell_i = q(n) \) for all \( i \). Let \( \text{Overlap}_i \) denote the event that the sequence \( \text{ctr}_i + 1, \ldots, \text{ctr}_i + q(n) \) overlaps with \( \text{ctr}^* + 1, \ldots, \text{ctr}^* + q(n) \) and let \( \text{Overlap} \) denote the event that \( \text{Overlap}_i \) occurs from some \( i \). By the union bound

\[
\Pr[\text{Overlap}] \leq \sum_{i=1}^{q(n)} \Pr[\text{Overlap}_i].
\]
**Probability of overlaps**

Fixing ctr*, event Overlap_i occurs exactly when ctr_i satisfies

\[
ctr^* + 1 - q(n) \leq ctr_i \leq ctr^* + q(n) - 1.
\]

Since there are \(2q(n) - 1\) values of ctr_i for which Overlap_i can occur,

\[
Pr[Overlap_i] = \frac{2q(n) - 1}{2^n}.
\]

Combining this with the union bound given on the previous slide, we have

\[
Pr[Overlap] \leq \sum_{i=1}^{q(n)} \frac{2q(n) - 1}{2^n} = \frac{2q(n)^2}{2^n}.
\]

Since \(q\) is polynomial, \(2q(n)^2 / 2^n\) is negligible and we have

\[
Pr \left[ PrivK^\text{cpa}_{\mathcal{A},\Pi}(n) = 1 \right] \leq \frac{1}{2} + \text{negl}.
\]
Showing $\Pi$ is CPA-secure

The next step is to show that this implies the $\Pi$ is CPA-secure. That is for any probabilistic polynomial-time $\mathcal{A}$ there exists a negligible function $\text{negl}'$ such that

$$\Pr[\text{PrivK}_{\mathcal{A},\Pi}^{\text{cpa}}(n) = 1] \leq \frac{1}{2} + \text{negl}' .$$

This proof is very similar to the analogous step in the proof of Theorem 3.25 and is left as an exercise.

Block length and security

Remark. If an input to the block cipher is used more than once then security can be violated. Thus, it is not only the key length of a block cipher than is important, but also its block length

Example. Suppose we use a block cipher with block length 64-bits. Even if a completely random function with this block length is used, an adversary can achieve success with probability roughly $\frac{1}{2} + \frac{q^2}{2^{63}}$ in a chosen-plaintext attack with it makes $q$ queries, each $q$ blocks long.
Bad news for

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Bois</th>
<th>owes</th>
<th>Bullwinkle</th>
<th>$10,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ciphertext</td>
<td>tsrwmqp</td>
<td>wplsmaka</td>
<td>os02%sb@</td>
<td>gwom*bxz</td>
</tr>
<tr>
<td>Altered Ciphertext</td>
<td>os02%sb@</td>
<td>wplsmaka</td>
<td>tsrwmqp</td>
<td>gwom*bxz</td>
</tr>
<tr>
<td>Ciphertext</td>
<td>Bullwinkle</td>
<td>owes</td>
<td>Bois</td>
<td>$10,000</td>
</tr>
</tbody>
</table>

*Well yes, but not our job. Issues of message integrity or message authentication should be dealt with separately from encryption.

Security against chosen-ciphertext attacks (CCA)

The experiment is defined for any private-key encryption scheme \( \Pi = (\text{Gen}, \text{Enc}, \text{Dec}) \), any adversary \( \mathcal{A} \), and any value \( n \) for the security parameter: The CCA indistinguishability experiment \( \text{PrivK}^{\text{CCA}}_{\mathcal{A}, \Pi}(n) \):

1. A key \( k \) is generated by running \( \text{Gen}(1^n) \).
2. The adversary \( \mathcal{A} \) is given \( 1^n \) and oracle access to \( \text{Enc}_k(\cdot) \) and \( \text{Dec}_k(\cdot) \). It outputs a pair of messages \( m_0, m_1 \in \mathcal{M} \) of the same length.
3. A random bit \( b \leftarrow \{0, 1\} \) is chosen. A challenge ciphertext \( c \leftarrow \text{Enc}_k(m_b) \) is computed and given to \( \mathcal{A} \).
4. The adversary \( \mathcal{A} \) continues to have oracle access to \( \text{Enc}_k(\cdot) \) and \( \text{Dec}_k(\cdot) \), but is not allowed to query the latter on the challenge ciphertext. Eventually \( \mathcal{A} \) outputs a bit \( b' \).
5. The output of the experiment is defined to be 1 if \( b' = b \), and 0 otherwise. We write \( \text{PrivK}^{\text{CCA}}_{\mathcal{A}, \Pi}(n) = 1 \) if the output is 1 and in this case we say that \( \mathcal{A} \) succeeded.
**CCA-secure**

**Definition 3.30.** A private-key encryption scheme \( \Pi = (\text{Gen}, \text{Enc}, \text{Dec}) \) had\(^*\) **indistinguishable encryption under a chosen-ciphertext attack** if for all probabilistic polynomial-time adversaries \( \mathcal{A} \) there exists a negligible function \( \text{negl} \) such that

\[
\Pr[\text{PrivK}_{\mathcal{A},\Pi}^\text{cca}(n) = 1] \leq \frac{1}{2} + \text{negl}(n),
\]

where the probability is taken over the random coins used by \( \mathcal{A} \), as well as the random coins used by the experiment (for choosing the key, the random bit \( b \), and any random coins used in the encryption process).

*Is such an attack feasible?*

**Insecurity of the schemes we have studied\(^*\)**

**Very bad news.** None of the schemes we have studied are CCA-secure.

**For example.** Consider Construction 3.25, where encryption is carried out as \( \text{Enc}_k(m) = \langle r, F_k(r) \oplus m \rangle \).

An adversary \( \mathcal{A} \) running in the CCA indistinguishability experiment can choose \( m_0 = 0^n \) and \( m_1 = 1^n \). Upon receiving \( c = \langle r, s \rangle \), the adversary can flip the first bit of \( s \) and ask for a decryption of the resulting ciphertext \( c' \). Since \( c' \neq c \) this is allowed, and the decryption oracle answers with either \( 10^{n-1} \) (in which case \( b = 0 \)) or \( 01^{n-1} \) (in which case \( b = 1 \)).

*Any encryption scheme that allows ciphertext to be manipulated in any "logical way" cannot be CCA-secure. More soon.*