Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 03, part 1
Defining Secure Encryption
Lessons learned so far

Crypto Design Lesson One

- The key space must be large enough to make brute-force attacks impractical (cf. Shift Cipher)

Crypto Design Lesson Two

- Large key space is a necessary, but not sufficient condition for a secure encryption scheme (cf. Vigenère Cipher)

But what does secure actually mean?
In this lecture

- What do we mean by secure?
- How do we know when a scheme is secure?
- Can we prove that some encryption scheme is secure?
Three principles of modern cryptography

**Definitions**
- Precise, mathematical model and formal definition of what security means

**Assumptions**
- Clearly stated and unambiguous

**Proofs**
- *Prove security* and move away from design-break-patch
Defining secure encryption

<table>
<thead>
<tr>
<th>Security guarantee/goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ What we want to achieve (or what we want to prevent the attacker from achieving)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threat model</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ What (real-world) capabilities the attacker is assumed to have</td>
</tr>
</tbody>
</table>
A private-key encryption scheme is defined by a message space $\mathcal{M}$, (key space $\mathcal{K}$) and algorithms (Gen, Enc, Dec):

- **Gen** (key-generation algorithm): outputs $k \in \mathcal{K}$
- **Enc** (encryption algorithm): takes key $k$ and message $m \in \mathcal{M}$ as input; outputs ciphertext $c \leftarrow \text{Enc}_k(m)$
- **Dec** (decryption algorithm): takes key $k$ and ciphertext $c$ as input; outputs $m$ or ”error”: $m = \text{Dec}_k(c)$
Private-key encryption
Threat models for encryption

- Ciphertext-only attack
  - One ciphertext
  - Many ciphertexts
- Known-plaintext attack
- Chosen-plaintext attack
- Chosen-ciphertext attack
Goal of secure encryption?

- How would you define what it means for encryption scheme \((\text{Gen}, \text{Enc}, \text{Dec})\) over message space \(\mathcal{M}\) to be secure?
- ...against a (single) ciphertext-only attack
Secure encryption?

”Impossible for the attacker to learn the key”

- The key is a means to an end, not the end itself
- Necessary (to some extent) but not sufficient
- Easy to design an encryption scheme that hides the key completely, but is insecure
- Can design schemes where most of the key is leaked, but the scheme is still secure
Secure encryption?

”Impossible for the attacker to learn the plaintext from the ciphertext”

► What if the attacker learns 90% of the plaintext?
Secure encryption?

“Impossible for the attacker to learn any character of the plaintext from the ciphertext”

- What if the attacker is able to learn (other) partial information about the plaintext?
  - e.g. salary is greater than 75K
- What if the attacker guesses a character correctly?
The right definition

<table>
<thead>
<tr>
<th>Secure encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regardless of any <strong>prior information</strong> the attacker has about the plaintext, the ciphertext should leak <strong>no additional information</strong> about the plaintext</td>
</tr>
</tbody>
</table>

How to formalize? $\implies$ defining **perfect secrecy** (next slides!)
End

Reference: From Section 1.3 until the end of Chapter 1 of the book.