Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 9, part 2
Security Against Chosen-Ciphertext Attacks (CCA)
Summary

We described a scheme based on **PRF/block cipher** in a given **mode of operation**

- Solves OTP limitation 1 (key as long as the message)
- Solves OTP limitation 2 (key used only once)
- EAV-secure (single-message secrecy)
- CPA-secure (multiple message secrecy)
Summary

 Threat model: attacker observes multiple ciphertexts $c_i$
 Security goal: given $c_i$ attacker cannot derive any information on any $m_i$
Summary

Threat model: attacker observes multiple ciphertexts \( c_i \)

Security goal: given \( c_i \) attacker can not derive any information on any \( m_i \)

So far considering only passive, eavesdropping attackers
What about Active Attackers?

What if the attacker can be active?

- Interfering with the communication channel
- Sending information on the communication channel
- Modifying what is sent over the channel
- Injecting traffic on the channel
Adversary $A$ Interfering with the Channel

$\text{k} \xrightarrow{\text{c}} \text{c} \leftarrow \text{Enc}_k(m) \xrightarrow{\text{c}'} \text{c'} \rightarrow \text{Receiver} \overset{\text{Dec}_k(c')}{\rightarrow} \text{m'} := \text{Dec}_k(c')$

- In the new model we don’t assume that the ciphertext can reach the receiver **unchanged**
- $A$ is allowed to **modify** $c$ to $c'$ and forward $c'$ to the receiver
- Receiver decrypts $c'$ to $m' \neq m$ and has **no way of detecting** the modification
Malleability

Question

How to capture this new property of the scheme in the presence of active attackers?

Malleability (informal)

A scheme is **malleable** if it is possible to modify a ciphertext and thereby cause a **predictable change to the plaintext**

Malleability can be dangerous e.g. encrypted bank transactions, encrypted email, etc.
Malleability

Observe
All the encryption schemes we have seen so far are malleable!

Simplest example: the OTP.
Malleability of the OTP

Plaintext $m = (m_0 \ldots m_n)$ as a sequence of $n$ bits encrypted with $n$-bit key $k$

Attacker flips the last bit of the ciphertext $c$ from $c_n$ to $c'_n$

The modification causes **predictable change to the plaintext**

Namely, the last bit of $m$ is flipped from $m_n$ to $m'_n = m_n \oplus 1$
Malleability

Implication

Perfect secrecy does not imply non-malleability

► i.e. a perfectly secret scheme may still be malleable
Malleability

Malleability attacks exist on all the encryption schemes we have seen so far

- **OTP, POTP**
  - Attack described above
- **CTR, OFB, stream ciphers**
  - Same as OTP
- **ECB**
  - Generate new valid $c$ from combining previously observed $c_i$
- **CBC**
  - Bit flip in $c_i$ causes bit flip in $m_{i+1}$
Adversary $A$ Injecting Messages On the Channel

- A special case of the "interfering" attack
- $A$ impersonates the sender and injects its own ciphertext $c'$
- By forcing the receiver to decrypt $c'$, $A$ may learn (something about) $m'$ (or $m$)
Chosen-ciphertext Attacks (CCA)

**CCA**

Models settings in which the attacker can influence what gets decrypted, and observe the effects

**How to model?**

- Allow attacker to submit ciphertexts of its choice* to the receiver, and learn the corresponding plaintext
- **In addition** to being able to carry out a chosen-plaintext attack

* With one restriction, described later
CPA vs. CCA

- CPA: A interacts with the sender i.e. has access to encryption oracle
- CCA: A interacts with the receiver i.e. has access to decryption oracle
  - in addition to access to an encryption oracle

- CCA is a stronger notion than CPA
- CCA implies CPA
CCA-security

\[ \text{PrivK}^{\text{cca}}_{A, \Pi}(n) \]

Define a randomized experiment \( \text{PrivK}^{\text{cca}}_{A, \Pi}(n) \):

- **k** \( \leftarrow \) \( \text{Gen}(1^n) \)
- **A**\((1^n)\) interacts with an encryption oracle \( \text{Enc}_k(\cdot) \), and a decryption oracle \( \text{Dec}_k(\cdot) \), and then outputs \( m_0, m_1 \) of the same length
- \( b \leftarrow \{0, 1\} \), \( c \leftarrow \text{Enc}_k(m_b) \), give \( c \) to **A**
- **A** continues to interact with \( \text{Enc}_k(\cdot) \) and \( \text{Dec}_k(\cdot) \), but may not request decryption of \( c \)
- **A** outputs \( b' \); **A** succeeds if \( b = b' \), and experiment evaluates to 1 in this case
CCA-security

Π is secure against chosen-ciphertext attacks (CCA-secure) if for all PPT attackers A, there is a negligible function $\epsilon$ such that

$$\Pr[\text{Priv}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$
CCA and Malleability

Fact

CCA-security implies non-malleability

If a scheme is malleable, then it cannot be CCA-secure:
1. Modify the challenge $c$ to $c'$
2. Submit $c'$ to the decryption oracle to get $m'$
3. The modification of $c$ to $c'$ **predicatably** modifies $m$ to $m'$
4. From $m'$ revert back the modification to recover $m_0$ that produced $c$
Is the CCA Model too Strong?

In the definition of CCA-security, the attacker can obtain the decryption of *any ciphertext of its choice* (besides the challenge ciphertext)

- Is this realistic?

There are scenarios where:

- One bit about decrypted ciphertexts is leaked
- The scenario occurs in the real world
- It can be exploited to learn the entire plaintext
End

Reference: Section 3.7.1