Lecture: Security and MACs

Notes Feb 5th:

Topics covered: Security games, Security modes of operation, MAC’s

Topic one: One time security

- Models the case where encryption is used just once
- Weakest security of all 3 types (weak because we want to use a encryption more than once)

What defines it as a game?

1) Adversary chooses \( m_0, m_1 \)
2) Oracle (black box encryption function) encrypts one at random \( C_b \)
3) Adversary guesses if \( C_b \) encrypts \( m_0 \) or \( m_1 \)

- Encryption is one time secure if the adversary’s success rate is less than: \( \frac{1}{2} + \epsilon \text{(negligible)} \)

Example of negligible: Something that is negligible is something that decreases exponentially

Technical definition: A negligible function is something that is \( < 1/\text{poly (} \ell \) \)

- Sometimes we say that an adversary is polynomial bounded (e.g. \( a/(2^\ell) \rightarrow \) it does not matter what the value of a is because the denominator is growing exponentially)

Say an adversary can decrypt the last bit:

- \( M_0 = 00000000 \)
- \( M_1 = 00000001 \)
- Get back \( C_b \rightarrow \) last bit 0? If so \( m_0 = 0 \) otherwise \( m_1 \)

Now lets say they win 1% of the time:

- Probability of success = \( (0.1) \cdot (0.99) \cdot (1/2) \)
- \( = \frac{1}{100} + \frac{99}{200} \)
- \( = \frac{1}{2} + \frac{1}{200} \)
- Larger than 1 + \( \epsilon \rightarrow \) insecure

Topic two: Secure against chosen plain text attacks: (CPA- Secure)

Game:

1) Adversary chooses any message and receives encryption
2) Adversary chooses \( m_0, m_1 \)
3) Oracle encrypts one at random; \( C_b \)
4) Adversary chooses any message and receives encryption
5) Adversary guesses

Encryption box in this case can perform multiple encryptions

Model using encryption function more than once. (We also call this an adaptive attack because step 4 can use the information of step 3)

**Topic three: ECB mode**

ECB -> not CPA

1) Adversary chooses \( m_0, m_1 \)
2) Oracle gives \( c_b \)
3) Adversary asks to have \( m_0 \) encrypted

\[ \text{Enc} (m_0) = c_b \rightarrow \text{guess } m_0 \text{ otherwise guess } m \]

Theory: no deterministic encryption function is ever CPA-secure

**Topic four: CBC (c.f CTR)**

1) Adversary chooses \( m_0, m_1 \)
2) Oracle returns \( c_b \)
3) Adversary asks for encryption of \( m_0 \)

CBC -> CPA secure

**CBC: IV’s**

1) Random encryption
2) Public
3) Generate IV after getting the message (ignoring this then it is no longer CPA-secure)

**Model access to a decryption function:**

1) Adversary chooses any message and receives encryption or any cipher text and receives decryption
2) Adversary chooses \( m_0, m_1 \);
3) Oracle encrypts one at random; \( c_b \)
4) Adversary chooses any message and receives encryption or any cipher text except \( C_b \) and receives plaintext
5) Adversary guesses

CBC (c.f CTR) is not CCA-secure:

**Attack:**

1) Adversary chooses: \( m_0 = A B C D \rightarrow \text{block } m1 = W X Y Z \)
2) Get back $Cb = [ ] [ ] [ ] [ ] cb' = [ ] [ ] [ ]$ FIG X
3) a) Create $cb'$
   b) Ask for decryption

**Topic 5: Plaintext awareness**

CCA-secure: any CPA secure encryption scheme where it is infeasible to generate a valid cipher text is CCA-secure

- CCA $\rightarrow$ CPA $\rightarrow$ OTS
- ECB = (multiple blocks) Not OTS
- CBC, CFB = CPA and OTS secure but not CCA
- ENC + MAC = CCA, CPA and OTS secure
- Stream ciphers = not CCA and the rest depends on analysis

**Topic 6: Message authentication codes (MACs)**

$a \text{ (tay/code)} = MAC_k \text{ (key)}$
Security properties:
Confidentiality -> encryption
Message integrity -> MAC’s
MAC’s ~ = keyed error correcting code -> no error correction just detection
Attacking a MAC: forgery:
Security -> infeasible to forge a code/tag on message without the key

Two styles of MACS
1) Based on hash functions
2) Block cipher based

Building a MAC from a hash:

**Attempt 1:** keyed hash function
m, a = H (k || m)

⇒ Doesn’t work because of *length extension attacks*

**Attempt 2:**
H (m || k) -> length extension secure
-> H (m || k || k’)

**Problem:**
m-m’ collision
H (m) = H (m’)
H (m || k) = H (m || k) [Using Merkle damgard]