EE 720: An Introduction to Number Theory and Cryptography (Spring 2019)

Lecture 10 — February 4, 2019

Instructor: Saravanan Vijayakumaran *Scribe: Saravanan Vijayakumaran*

1 Lecture Plan

- Discuss the insecurity of chained CBC block cipher mode
- Define CCA-security
- Describe the padding oracle attack

2 Chained CBC Mode

- Chained CBC mode is a stateful variant of the CBC mode where the last block of the previous ciphertext is used as the IV when encrypting the next message.
- Chained CBC mode is not secure
- Consider the following adversary $\mathcal{A}$ in the $\texttt{PrivK}^{\texttt{cpa}}_{\mathcal{A},\Pi}(n)$ experiment.
  - $\mathcal{A}$ chooses two messages consisting of two n-bit blocks each: $\mathbf{m}_0 = (m_0^0, m_1^0)$ and $\mathbf{m}_1 = (m_0^1, m_1^1)$.
  - The experimenter chooses a bit b and encrypts $\mathbf{m}_b$ using chained CBC mode. The challenge ciphertext consists of three n-bit blocks (IV, c_1, c_2).
  - Now suppose the adversary queries the encryption oracle on message $c_2 \oplus IV \oplus m_0^0$. The encryption oracle will be using the ciphertext c_2 as the initial value to answer the query.
  - If b was 0, then the ciphertext c_3 returned from the encryption oracle will be equal to c_1. Thus the adversary can guess the bit b with a probability equal to 1 as long as $m_0^0 \neq m_0^1$.

3 Chosen-Ciphertext Attack Security

- Previously, we considered ciphertext-only attacks and chosen-plaintext attacks. Known-plaintext attacks are weaker than chosen-plaintext attacks, so an encryption scheme which is CPA-secure will also be KPA-secure.
- We now consider *chosen-ciphertext attacks*. Here, the adversary has access to a decryption oracle $\texttt{Dec}_k(\cdot)$ which decrypts ciphertexts chosen by the adversary. The adversary is not allowed to send the ciphertext exchanged between the honest parties to the decryption oracle.
- For a formal definition of the CCA threat model, consider the *CCA indistinguishability experiment* $\texttt{PrivK}^{\texttt{cca}}_{\mathcal{A},\Pi}(n)$:

1. A key k is generated by running $\texttt{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given 1^n and oracle access to $\texttt{Enc}_k(\cdot)$ and $\texttt{Dec}_k(\cdot)$. It outputs a pair of messages $m_0, m_1 \in \mathcal{M}$ with $|m_0| = |m_1|$.
3. A uniform bit $b \in \{0,1\}$ is chosen. Ciphertext $c \leftarrow \texttt{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$. c is called the *challenge ciphertext.*
4. The adversary $\mathcal{A}$ continues to have oracle access to $\texttt{Enc}_k(\cdot)$ and $\texttt{Dec}_k(\cdot)$, but is not allowed to query the latter on the challenge ciphertext itself. Eventually, A outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. If output is 1, we say that $\mathcal{A}$ succeeds.

Definition. *A private-key encryption scheme $\Pi = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ has* ***indistinguishable encryptions under a chosen-ciphertext attack****, or is* ***CCA-secure****, if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there is a negligible function $\mathtt{negl}$ such that, for all n,*

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

- None of the encryption schemes we have seen so far is CCA-secure. Consider the CPA-secure scheme where $\texttt{Enc}_k(m) = \langle r, F_k(r) \oplus m\rangle$. Consider the following adversary $\mathcal{A}$ in the CCA indistinguishability experiment.

  1. $\mathcal{A}$ chooses $m_0 = 0^n$ and $m_1 = 1^n$.
  2. Upon receiving the challenge ciphertext $c = \langle r, s\rangle = \langle r, F_k(r) \oplus m_b\rangle$, $\mathcal{A}$ asks for the decryption of $c' = \langle r, s'\rangle = \langle r, s \oplus 10^{n-1}\rangle$ i.e. the bit $n+1$ in c is flipped.
  3. The oracle answers with $m' = s' \oplus F_k(r) = F_k(r) \oplus m_b \oplus 10^{n-1} \oplus F_k(r) = m_b \oplus 10^{n-1}$.
  4. m' is 10^{n-1} if $b = 0$ and 01^{n-1} if $b = 1$. So the adversary succeeds with probability 1.

4 Padding Oracle Attack

- Do chosen-ciphertext attacks model any real-world attack? The answer is yes. Padding oracle attacks are one such example.

- Recall the CBC block cipher mode used encrypt plaintext whose length is longer than the block length of a block cipher.

  – Let $m = m_1, m_2, \ldots, m_l$ where $m_i \in \{0,1\}^n$.
  – Let F be a length-preserving block cipher with block length n.
  – A uniform *initialization vector (IV)* of length n is first chosen.
  – $c_0 = IV$. For $i = 1, \ldots, l$, $c_i := F_k(c_{i-1} \oplus m_i)$.
  – For $i = 1, 2, \ldots, l$, $m_i := F_k^{-1}(c_i) \oplus c_{i-1}$.

- The above scheme assumes that the plaintext length is a multiple of n. The plaintext is usually *padded* to satisfy this constraint. For convenience we will refer to the original plaintext as the *message* and the result after padding as the *encoded data*.

- A popular padding scheme is the PKCS #5 padding.

- Assume that the original message m has an integral number of bytes. Let L be the blocklength of the block cipher in bytes.
- Let b denote the number of bytes required to be appended to the original message to make the encoded data have length which is a multiple of L. Here, b is an integer from 1 to L ($b = 0$ is not allowed).
- We append to the message the integer b (represented in 1 byte) repeated b times. For example, if 4 bytes are needed then the `0x04040404` is appended. Note that L needs to be less than 256. Also, if the message length is already a multiple of L, then L bytes need to be appended each of which is equal to L.

- The encoded data is encrypted using CBC mode. When decrypting, the receiver first applies CBC mode decryption and then checks that the encoded data is correctly padded. The value b of the last byte is read and then the final b bytes of the encoded data is checked to be equal to b.
- If the padding is incorrect, the standard procedure is to return a "bad padding" error. The presence of such an error message provides the an adversary with a *partial* decryption oracle. While this may seem like meaningless information, it enables the adversary *to completely recover the original message for any ciphertext of its choice.*
- See pages 99–100 for a complete description of the attack.
- One solution is to use message authentication codes.

5 References and Additional Reading

- Section 3.7 from Katz/Lindell