HomeWork2

Jonathan Degrange

23th September 2024

Problem 1: PRF Constructions

1. $F'_k(x) = F_k(\overline{x}) \| F_k(x)$

Answer: In this construction, $F'_k(x)$ concatenates the PRF applied to the bitwise negation $\overline{x}$ with the PRF applied to x. Despite that $F_k(x)$ and $F_k(\overline{x})$ are individually secure, the relationship between x and $\overline{x}$ is deterministic. The attacker can analyse the link between $F_k(x)$ and $F_k(\overline{x})$ and find some informations about $F_k(x)$ and the key k.

So this function cn be dinstinguished from a random function. It is then not secure

2. $F''_k(x) = F_{k_1}(x) \oplus F_{k_2}(x) \| x$

Answer: In this construction, we XOR the output of two PRFs with two different key $k1$ and $k2$. This output should behave as a PRF yet. But we also concatenate to this output the value x. This can give some information to the attacker about x, and be a weakness and a lack of security. We then cannot say that $F''_k(x)$ is undistinguishable from a random function. This funciton is not secure.

3. $F'''_k(x) = lsb(F_{k_1}(x)) \| F_{k_2}(x)$

Answer: In this construction, the least significant bit of $F_{k_1}(x)$ is concatenated with the output of $F_{k_2}(x)$. While $F_{k_2}(x)$ is still a secure PRF, truncating $F_{k_1}(x)$ to only its LSB could lead to a loss of entropy and security. Then, the lsb may not be enough to give us a funciton undistinguishable from random. It could also give some informations about the pattern, exploitable by an attacker.

This function is not secure.

Problem 2

Construction 1

Encryption:

$$E_k(m) = (RH(y), G(RH(y)) \oplus m)$$

with $y = F_k(r)$, and $RH(y)$ is the right half of y.

Decryption:

1. Use $RH(y)$ to compute $G(RH(y))$.
2. Recover the message m as: $m = G(RH(y)) \oplus c$.

Security: Secure against CPA because a fresh random r makes each encryption unique, and G and F_k are pseudorandom.

Construction 2

Encryption:

$$E_k(m) = (r, F_k(F_k(r)) \oplus m)$$

Decryption:

1. Compute $F_k(F_k(r))$.
2. Recover the message m as: $m = F_k(F_k(r)) \oplus c$.

Security: Secure against CPA due to random r, ensuring each encryption is different for the same message, with strong pseudorandomness from F_k.

Construction 3

Encryption:

$$E_k(m) = (r, r', F_k(1^n) \oplus m_1, F_k(r) \oplus m_2, F_k(r') \oplus m_3)$$

Decryption:

1. Compute $F_k(1^n)$, $F_k(r)$, and $F_k(r')$.
2. Recover m_1, m_2, m_3 as: $m_1 = F_k(1^n) \oplus c_1$, $m_2 = F_k(r) \oplus c_2$, $m_3 = F_k(r') \oplus c_3$.

Security: Secure against CPA due to random r and r', with pseudorandom outputs from F_k, ensuring different ciphertexts for the same plaintext.

Problem 3: OTP and Feistel Network

1. Alice's Claim on OTP

Alice claims OTP is deterministic and therefore not secure against CPA due to the lack of randomness.

Analysis: Alice is *wrong*. OTP is not deterministic. The key is random and as long as the message, making the ciphertext $c = m \oplus k$ different every time. OTP is perfectly secure if the key is used only once and is truly random.

Conclusion: Alice's claim is false. OTP is secure against CPA as long as the key is random, unique, and never reused.

2. Decrypting the Feistel Network

Given $g_k(m)$ from a Feistel network, we can reverse the process to recover m.

Decryption Process: In a Feistel network, for each round:

$$L_{i+1} = R_i, \quad R_{i+1} = L_i \oplus F_k(R_i)$$

Decryption works by reversing the steps:

$$R_i = L_{i+1}, \quad L_i = R_{i+1} \oplus F_k(L_{i+1})$$

Starting with (L_n, R_n), reverse the operations to get (L_0, R_0), which is the original message m.

Conclusion: Yes, we can decrypt the Feistel network by reversing the rounds.