

CSE 5014: Cryptography and Network Security
2024 Spring Semester Written Assignment # 3
Due: May 7th, 2024, please submit at the beginning of class
Sample Solutions

Q.1 Let $\Pi = (Gen, Enc, Dec)$ be the CTR mode encryption scheme and $\tilde{\Pi} = (\tilde{Gen}, \tilde{Enc}, \tilde{Dec})$ be the encryption scheme obtained from Π by using a *truly random* function f instead of a pseudorandom function F_k (This is to say that \tilde{Gen} picks uniformly $f \in Func_n$, while Gen picks uniformly $k \in \{0, 1\}^n$, and \tilde{Enc} uses f where Enc uses F_k). Show that there is a negligible function $negl$, such that for any PPT adversary A , it holds that

$$\left| \Pr[PrivK_{A,\Pi}^{CPA}(n) = 1] - \Pr[PrivK_{A,\tilde{\Pi}}^{CPA}(n) = 1] \right| \leq negl(n).$$

Solution:

Assume the statement is false, that is, for some PPT adversary A ,

$$\left| \Pr[PrivK_{A,\Pi}^{CPA}(n) = 1] - \Pr[PrivK_{A,\tilde{\Pi}}^{CPA}(n) = 1] \right| > t(n)$$

for some non-negligible $t(n)$. We want to show how to construct a PPT distinguisher D contradicting the requirement from the definition of pseudorandom functions, i.e., it should hold that

$$\left| \Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1] \right| > t(n).$$

We define the working of the distinguisher D as follows, where D is given access to some oracle $\mathcal{O}: \{0, 1\}^n \rightarrow \{0, 1\}^n$ and receives an input 1^n .

- (1) Run $A(1^n)$, and when $A(1^n)$ queries its oracle to the encryption function for the i -th time with a message made up from ℓ_i message blocks m_1, \dots, m_{ℓ_i} , do the following
 - (a) Choose a uniform initial value $ctr_i \in \{0, 1\}^m$
 - (b) Query \mathcal{O} for $j = 1, \dots, \ell_i$ to obtain $y_j := \mathcal{O}(ctr_i + j)$
 - (c) Return the ciphertext blocks $\langle ctr_i, c_1, \dots, c_{\ell_i} \rangle := \langle ctr_i, y_1 \oplus m_1, \dots, y_{\ell_i} \oplus m_{\ell_i} \rangle$ to A

- (2) Once A outputs the messages m_0, m_1 consisting of ℓ^* blocks $m_{0,1}, \dots, m_{0,\ell^*}, m_{1,1}, \dots, m_{1,\ell^*}$, respectively, choose a uniform bit $b \in \{0, 1\}$ and do the following:
- (a) Choose a uniform initial value $ctr^* \in \{0, 1\}^m$
 - (b) Query \mathcal{O} for $j = 1, \dots, \ell^*$ to obtain $y_j^* := \mathcal{O}(ctr^* + j)$ Return the challenge ciphertext blocks $\langle ctr^*, c_1^*, \dots, c_{\ell^*}^* \rangle := \langle ctr^*, y_1 \oplus m_{b,1}, \dots, y_{\ell^*} \oplus m_{b,\ell^*} \rangle$ to A
- (3) Answer queries to the encryption oracle as above, until A produces an output bit b' . Then output 1 if $b = b'$, and 0 otherwise.

We first argue that D is PPT: Each of the above steps clearly only incurs polynomial overhead for each of the oracle calls from A , and as a PPT adversary, A may only pose a polynomial number of queries to the encryption oracle and may itself only run in polynomial time, hence D also runs in polynomial time.

Note that D is essentially just the experiment $PrivK_{A,\Pi}^{CPA}$ or $PrivK_{A,\tilde{\Pi}}^{CPA}$, depending on which oracle D is given, implemented as an algorithm, where the oracle queries of A are spelled out step-by-step, with the first step of key generation in the experiment being simulated by uniformly choosing $k \in \{0, 1\}^n$ or $f \in Func_n$, respectively. This is equivalent since this is also how Gen and \tilde{Gen} generate the keys by definition of CTR . Therefore, by definition of Π and $\tilde{\Pi}$, we have that $D^{F_k(\cdot)}(1^n)$ and $PrivK_{A,\Pi}^{CPA}(n)$ are identically distributed, and $D^{f(\cdot)}(1^n)$ and $PrivK_{A,\tilde{\Pi}}^{CPA}(n)$ are identically distributed, for uniformly chosen f . In other words,

$$\begin{aligned} \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k(\cdot)}(1^n) = 1] &= \Pr [PrivK_{A,\Pi}^{CPA}(n) = 1], \\ \Pr_{f \leftarrow Func_n} [D^{f(\cdot)}(1^n) = 1] &= \Pr [PrivK_{A,\tilde{\Pi}}^{CPA}(n) = 1]. \end{aligned}$$

Thus, (2) follows directly from (1).

□

Q.2 Let F_k be a PRF and G be a PRG with expansion factor $n \mapsto n + 1$. For each of the following encryption schemes, decide whether the scheme has indistinguishable encryptions in the presence of an eavesdropper and whether it is CPA-secure. (In each case, the shared key is a uniform $k \in \{0, 1\}^n$.) Explain your answer.

- (a) To encrypt $m \in \{0, 1\}^n$, output the ciphertext $m \oplus F_k(0^n)$.
- (b) The one-time pad.
- (c) To encrypt $m \in \{0, 1\}^{n+1}$, choose uniform $r \in \{0, 1\}^n$ and output the ciphertext $\langle r, G(r) \oplus m \rangle$.
- (d) To encrypt $m \in \{0, 1\}^{2n}$, parse m as $m_1 || m_2$ with $|m_1| = |m_2| = n$, then choose uniform $r \in \{0, 1\}^n$ and output the ciphertext $\langle r, m_1 \oplus F_k(r), m_2 \oplus F_k(r + 1) \rangle$.

Solution:

- (a) This scheme has indistinguishable encryptions in the presence of an eavesdropper. To see this, we observe that $F_k(0^n)$ is pseudorandom. The formal proof is similar to the proof of Theorem 3.18 in the textbook. The scheme is not CPA-secure as encryption is deterministic.
- (b) The one-time pad is perfectly indistinguishable which immediately implies that it has also indistinguishable encryption in the presence of an eavesdropper. It is not CPA-secure as encryption is deterministic.
- (c) This scheme does not even have indistinguishable encryptions in the presence of an eavesdropper, as decryption can be done in polynomial time without knowing the key: On input $(r, G(r) \oplus m)$, we just compute $G(r)$ and then output $G(r) \oplus (G(r) \oplus m) = m$. Therefore, it cannot be CPA-secure as well.
- (d) This scheme is CPA-secure. The proof is similar to the correctness proof of CTR mode with just two blocks. Therefore, it has indistinguishable encryptions in the presence of an eavesdropper as well.

□

Q.3 Let F_k be a PRF. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. (In each case Gen outputs a uniform $k \in \{0, 1\}^n$ and $[i]_2$ denotes the $\frac{n}{2}$ -bit binary encoding of i).

- (a) To authenticate a message $m = m_1 \dots m_\ell$, where $m_i \in \{0, 1\}^n$, compute

$$t := F_k(m_1) \oplus \dots \oplus F_k(m_\ell).$$

- (b) To authenticate a message $m = m_1 \cdots m_\ell$, where $m_i \in \{0, 1\}^{\frac{n}{2}}$, compute

$$t := F_k([1]_2 || m_1) \oplus \cdots \oplus F_k([\ell]_2 || m_\ell).$$

- (c) To authenticate a message $m = m_1 \cdots m_\ell$, where $m_i \in \{0, 1\}^{\frac{n}{2}}$, choose uniform $r \leftarrow \{0, 1\}^n$, and compute

$$t := (r, F_k(r) \oplus F_k([1]_2 || m_1) \oplus \cdots \oplus F_k([\ell]_2 || m_\ell)).$$

Solution:

We construct an adversary A for each of the MACs.

- (a) On input 1^n , A queries $(0^n 1^n)$ and gets $t = \text{Mac}_k(0^n 1^n) = F_k(0^n) \oplus F_k(1^n)$. Now A outputs $(1^n 0^n, t)$. This is a valid message-tag pair as $\text{Mac}_k(1^n 0^n) = F_k(1^n) \oplus F_k(0^n) = F_k(0^n) \oplus F_k(1^n) = t$, i.e., A wins with probability 1.
- (b) On input 1^n , A queries $m_0 = 0^n$, $m_1 = 0^{\frac{n}{2}} 1^{\frac{n}{2}}$ and $m_2 = 1^n$. We denote the tags as t_0, t_1 and t_2 . Now it holds that

$$\begin{aligned} t_0 \oplus t_1 \oplus t_2 &= (F_k([1] || 0^{\frac{n}{2}}) \oplus F_k([2] || 0^{\frac{n}{2}})) \oplus (F_k([1] || 0^{\frac{n}{2}}) \oplus F_k([2] || 1^{\frac{n}{2}})) \\ &\quad \oplus (F_k([1] || 1^{\frac{n}{2}}) \oplus F_k([2] || 1^{\frac{n}{2}})) \\ &= F_k([2] || 0^{\frac{n}{2}}) \oplus F_k([1] || 1^{\frac{n}{2}}) \\ &= F_k([1] || 1^{\frac{n}{2}}) \oplus F_k([2] || 0^{\frac{n}{2}}) \\ &= \text{Mac}_k(1^{\frac{n}{2}} 0^{\frac{n}{2}}) \end{aligned}$$

Therefore, A outputs $(1^{\frac{n}{2}} 0^{\frac{n}{2}}, t_0 \oplus t_1 \oplus t_2)$ and wins with probability 1.

- (c) Let $m \in \{0, 1\}^{\frac{n}{2}}$ be an arbitrary message. Then A outputs $(m, ([1]_2 || m))$. This is a valid message-tag pair as Mac_k could choose $r = [1]_2 || m$ and output

$$t = (r, F_k(r) \oplus F_k([1]_2 || m)) = (r, 0^n)$$

Consequently, A wins with probability 1.

□

Q.4 Let F_k be a PRF. Show that the following MAC for messages of length $2n$ is *insecure*: Gen outputs a uniform $k \in \{0, 1\}^n$. To authenticate a message $m_1 || m_2$ with $|m_1| = |m_2| = n$, compute the tag $F_k(m_1) || F_k(F_k(m_2))$

Solution: Let $m^1 = m_1 || m_1$, then $t^1 = t_1^1 || t_2^1 = F_k(m_1) || F_k(F_k(m_1))$; let $m^2 = m_2 || m_2$, then $t^2 = t_1^2 || t_2^2 = F_k(m_2) || F_k(F_k(m_2))$. Hence, for the message $m = m_1 || m_2$,

$$t = t_1^1 || t_2^2.$$

□

Q.5 Prove the following modifications of basic CBC-MAC do not yield a secure MAC (even for fixed-length messages).

- (1) Mac outputs all blocks t_1, \dots, t_ℓ , rather than just t_ℓ . (Verification only checks whether t_ℓ is correct.)
- (2) A random initial block is used each time a message is authenticated. That is, choose uniform $t_0 \in \{0, 1\}^n$, run basic CBC-MAC over the “message” t_0, m_1, \dots, m_ℓ , and output the tag $\langle t_0, t_\ell \rangle$. Verification is done in the natural way.

Solution:

- (1) The attacker queries $m^1 = B_0 || B_1$, and gets the tag $t^1 = t_0 || t_1$, where $t_0 = F_k(B_0)$; the attacker queries $m^2 = B_2 || B_3$, and gets the tag $t^2 = t_2 || t_3$ where $t_2 = F_k(B_2)$. Hence, we have

$$MAC_k(B_0 || B_2^*) = F_k(B_0) || F_k(F_k(B_0) \oplus B_2^*) = t_0 || F_k(t_0 \oplus B_2^*).$$

Let $t_0 \oplus B_2^* = B_2$, i.e., $B_2^* = t_0 \oplus B_2$. Then we have

$$MAC_k(B_0 || t_0 \oplus B_2) = t_0 || F_k(t_0 \oplus t_0 \oplus B_2) = t_0 || F_k(B_2) = t_0 || t_2.$$

This means that $\langle B_0 || t_0 \oplus B_2, t_0 || t_2 \rangle$ is a valid pair of message and tag.

- (2) The attacker queries $m^1 = B_0 || B_1$, and gets the tag $t^1 = \langle r_1, t_1 \rangle$; the attacker queries $m^2 = B_2 || B_3$, and gets the tag $t^2 = \langle r_2, t_2 \rangle$. Hence, we have for the message $m^* = B_0 || B_1 || t_2 \oplus r_2 || B_2 || B_3$, $t^* = \langle r, t_2 \rangle$ should be a valid tag.

□

Q.6 Suppose that the Shenzhen Traffic Police Department comes up with a new license plate with a special serial number format. This serial number format consists of only 3 letters: first two being a digit (0 to 9) and last one being an English uppercase letter (A to Z). Each serial number is randomly generated when issued.

- (1) Suppose that Alica and Bob apply for this new license plate. What is the probability that both of them receive the same plate number?
- (2) Suppose that the Shenzhen Traffic Police Department wants to ensure that the probability that at least two license plates have the same number is less than 1%. What is the maximum number of this type of license plates that they can issue?
- (3) Suppose that the Shenzhen Traffic Police Department wants to issue exactly 50 license plates. How many more DIGITs should be added at the end of this serial number format in order to ensure that the probability that at least two license plates have the same number is still less than 1%?

Solution:

- (1) There are $10^2 * 26 = 2600$ combinations of license plate numbers. We have

$$\begin{aligned}
 & \Pr[\text{both get the same number}] \\
 &= \Pr[\text{both get } 00A] + \Pr[\text{both get } 00B] + \cdots + \Pr[\text{both get } 99Z] \\
 &= (1/2600) * (1/2600) + (1/2600) * (1/2600) + \cdots + (1/2600) * (1/2600) \\
 &= 2600 * (1/2600) * (1/2600) \\
 &= 1/2600.
 \end{aligned}$$

- (2) Recall the probability of no-collision is $P_0 \approx e^{k(1-k)/2n}$. In this case, $n = 2600$ and $P_0 \geq 0.99$. Thus, we can solve for k based on this equation. After solving it, we get $k \leq 7$. The answer is 7 license plates.

- (3) It assumes that $k = 50$ and $P_0 \geq 0.99$ and we want to solve for n . Following the same equation, we get $n > k * (1 - k) / (2 * \log P_0)$. After solving it, we get $n \geq 121887$, Clearly, at least two digits need to be added.

□

Q.7 Let (Gen, H) be a collision resistant hash function and define the hash function $\tilde{H} = (Gen, \tilde{H})$ such that

$$\tilde{H}^s(x) := H^s(H^s(x)).$$

Prove or disprove: \tilde{H} is a collision resistant hash function, where s is the key obtained by Gen .

Solution:

\tilde{H} is a collision resistant hash function. We will prove this by reduction, i.e., we assume that \tilde{H} is not collision resistant and show that this would imply that Π is not collision resistant.

If \tilde{H} is not collision resistant, then there is a PPT adversary \tilde{A} such that

$$\Pr[\text{Hash-col}_{\tilde{A}, \tilde{\Pi}}(n) = 1] \geq \frac{1}{q(n)}$$

We use \tilde{A} to construct A as follows: On input s , A simulates \tilde{A} . The latter will output x, x' eventually. Now A checks whether $\tilde{H}^s(x) = \tilde{H}^s(x')$ and $x \neq x'$. If this is not the case A will just output x and x' (we do not care about this case). Otherwise A checks whether $H^s(x) = H^s(x')$. If this is the case, then a collision was found and A outputs x and x' . Otherwise we know that $H^s(x) \neq H^s(x')$ and $H^s(H^s(x)) = H^s(H^s(x'))$, that is, a collision is found, too. A outputs $H^s(x)$ and $H^s(x')$ in this case.

For the analysis let $\text{Succ}_{\tilde{A}}(n)$ be the event that \tilde{A} finds a collision in the execution of the Hash-col experiment with adversary \tilde{A} on input n . Clearly, we have

$$\Pr[\text{Succ}_{\tilde{A}}(n)] = \Pr[\text{Hash-col}_{\tilde{A}, \tilde{\Pi}}(n) = 1] > \frac{1}{q(n)}.$$

Furthermore the case analysis above shows that

$$\Pr[\text{Hash-col}_{A, \Pi}(n) = 1 | \text{Succ}_{\tilde{A}}(n)] = 1.$$

Putting everything together and by applying the law of total probability we get

$$\begin{aligned}
& \Pr[\text{Hash} - \text{col}_{A,\Pi}(n) = 1] \\
&= \Pr[\text{Hash} - \text{col}_{A,\Pi}(n) = 1 | \text{Succ}_{\tilde{A}}(n)] \cdot \Pr[\text{Succ}_{\tilde{A}}(n)] \\
&\quad + \Pr[\text{Hash} - \text{col}_{A,\Pi}(n) = 1 | \neg \text{Succ}_{\tilde{A}}(n)] \cdot \Pr[\neg \text{Succ}_{\tilde{A}}(n)] \\
&\geq \Pr[\text{Hash} - \text{col}_{A,\Pi}(n) = 1 | \text{Succ}_{\tilde{A}}(n)] \cdot \Pr[\text{Succ}_{\tilde{A}}(n)] \\
&= 1 \cdot \Pr[\text{Succ}_{\tilde{A}}(n)] \\
&> \frac{1}{q(n)}.
\end{aligned}$$

This completes the reduction as Π is a collision resistant hash function. Therefore our assumption was wrong and $\tilde{\Pi}$ is indeed collision resistant.

□

Q.8 Define $DESY_{k,k'}(x) = DES_k(x \oplus k')$. The key length of $DESY$ is 120 bits, which is the sum of the lengths of k and k' . Show a key-recovery attack on $DESY$ taking time and space approximately 2^{64} .

Solution: The observation is that an optimized meet-in-the-middle attack can be used here. Let $\{(x_i, y_i)\}$ be a set of plaintext/ciphertext pairs. For every $k \in \{0, 1\}^{56}$ compute $z_1 = DES_k^{-1}(y_1)$. Since $y_1 = DES_k(x_1 \oplus k')$ we have that if k is the correct first half of the key, then $k' = x_1 \oplus z_1$. That is, for every guess of k , we obtain a single possibility for k' . The other plaintext/ciphertext pairs can then be used to check whether this candidate key (k, k') is correct.

This attack requires time 2^{56} and constant space.

□

Q.9 Let $Feistel_{f_1, f_2}(\cdot)$ denote a two-round Feistel network using functions f_1 and f_2 (in that order). Show that if $Feistel_{f_1, f_2}(L_0, R_0) = (L_2, R_2)$, then $Feistel_{f_2, f_1}(R_2, L_2) = (R_0, L_0)$.

Solution: The formula for computing $Feistel_{f_1, f_2}(L_0, R_0)$ is:

$$\begin{aligned} L_1 &= R_0 \\ R_1 &= L_0 \oplus f_1(R_0) \\ L_2 &= R_1 = L_0 \oplus f_1(R_0) \quad (*) \\ R_2 &= L_1 \oplus f_2(R_1) = R_0 \oplus f_2(L_2) \quad (**). \end{aligned}$$

Now, taking this pair (L_2, R_2) , we compute $Feistel_{f_2, f_1}(R_2, L_2)$. We denote $L'_0 = R_2$ and $R'_0 = L_2$ and the intermediate values computed by L'_1, R'_1 and L'_2, R'_2 . Using the same formula as above, we have:

$$\begin{aligned} L'_1 &= R'_0 \\ R'_1 &= L'_0 \oplus f_2(R'_0) \\ L'_2 &= L'_0 \oplus f_2(R'_0) \\ R'_2 &= L'_1 \oplus f_1(R'_1) = R'_0 \oplus f_1(L'_2) \end{aligned}$$

(note that f_2 and f_1 have been reversed). Plugging in $L'_0 = R_2$ and $R'_0 = L_2$, we have

$$L'_2 = R_2 \oplus f_2(L_2) \text{ and } R'_2 = L_2 \oplus f_1(L'_2).$$

Using Eqn. (**), we have $R_2 \oplus f_2(L_2) = R_0$ and thus $L'_2 = R_0$. Thus, $R'_2 = L_2 \oplus f_1(L'_2) = L_2 \oplus f_1(R_0)$. However, by Eqn. (*), we have $L_2 \oplus f_1(R_0) = L_0$ and so $R'_2 = L_0$. We then conclude that $Feistel_{f_2, f_1}(R_2, L_2) = (R_0, L_0)$, as required.

□