CSE 5014: Cryptography and Network Security
2023 Spring Semester Written Assignment # 3
Due: May 9th, 2023, please submit at the beginning of class

Sample Solutions

Q.1 Let G be a pseudorandom generator (PRG) where $|G(s)| > 2 \cdot |s|$.

1. Define $G'(s) := G(s||0^{|s|})$. Is G' necessarily a pseudorandom generator?
2. Define $G'(s) := s_1||G(s_2)$, where $s = s_1||s_2$ and $s_1, s_2 \in \{0,1\}^{|s|/2}$. Is G' necessarily a pseudorandom generator?

Solution:

1. G' is *not* necessarily a PRG. Let $\hat{G}$ be any PRG that has stretch $\ell(n) > 4n$, and define $G(s) = G(s_1||s_2) = \hat{G}(s_2)$, where $|s_1| = |s_2| = |s|/2$. First, note that G is length doubling since $|\hat{G}(s_2)| = 4 \cdot |s|/2 = 2|s|$. Next, we prove that G is a PRG. Assume to the contrary that there is a polynomial-time algorithm A and a polynomial p such that for infinitely many n's,

$$|\Pr[A(G(U_n)) = 1] - \Pr[D(U_{2n}) = 1]| \geq \frac{1}{p(n)}.$$

Then, we construct a polynomial-time algorithm $\hat{A}$ for the generator $\hat{G}$. Upon input a string $r \in \{0,1\}^{2n}$, algorithm $\hat{A}$ invokes A upon input r and outputs whatever A does. We have that

$$\Pr[\hat{A}(\hat{G}(U_{n/2})) = 1] = \Pr[A(G(U_n)) = 1],$$

and

$$\Pr[\hat{A}(U_{2n}) = 1] = \Pr[A(U_{2n}) = 1].$$

Therefore, for infinitely many n's we have that

$$|\Pr[\hat{A}(\hat{G}(U_{n/2})) = 1] - \Pr\hat{A}(U_{2n}) = 1]| \geq \frac{1}{p(n)},$$

contradicting to the pseudorandomness of $\hat{G}$. We have now proved that G is a PRG, it remains to show that G' is not. However, for every s, $G'(s) = G(s||0^{|s|}) = \hat{G}(0^{|s|})$. Therefore, G' can easily be distinguished from random by computing $\hat{G}(0^{|s|})$ and comparing it to the input.

2. G' is a PRG. We prove this by contradiction: if we can distinguish a string generated by G' from a random string, then we will also be able to distinguish one generated by G from a random string as well.

   Assume that the stretch of the PRG G is $\ell(n)$. Then G' outputs a string of length $n/2 + \ell(n/2)$. Assume that A' is a polynomial-time distinguisher such that there is a polynomial p, for infinitely many n's,

$$|\Pr[A'(r) = 1] - \Pr[A'(G'(s)) = 1]| \geq \frac{1}{p(n)},$$

   where r is chosen uniformly at random form $\{0,1\}^{n/2+\ell(n/2)}$ and s is chosen uniformly at random form $\{0,1\}^n$. Now we construct a polynomial-time algorithm A: upon input a string $r \in \{0,1\}^{n/2}$, A invokes A' upon input r and outputs whatever A' does. We have that

$$\Pr[A(G(U_{n/2})) = 1] = \Pr[A'(G'(U_n)) = 1],$$

   and

$$\Pr[A(U_{\ell(n/2)}) = 1] = \Pr[A'(U_{n/2+\ell(n/2)}) = 1].$$

   Therefore, for infinitely many n's we have that

$$|\Pr[A(G(U_{n/2})) = 1] - \Pr[A(U_{\ell(n/2)}) = 1]| \geq \frac{1}{p(n)},$$

   contradicting to the fact that G is a PRG. Thus, G' must be a PRG.

□

Q.2 Consider the following keyed function F: for the security parameter n, the key is a matrix $A \in \mathbb{M}(n \times n, \mathbb{F}_2)$ and a vector $b \in \mathbb{F}_2^n$, where $\mathbb{F}_2$ denotes the field with 2 elements, i.e., $\mathbb{F}_2 = (\{0,1\}, \oplus, \cdot)$ and $\mathbb{F}_2^n$ denotes the corresponding vector space of dimension n. Now we define $F_{A,b} := \mathbb{F}_2^n \to \mathbb{F}_2^n$ by

$$F_{A,b} = Ax + b.$$

Decide whether F is a PRF and prove your answer.

Solution: F is *not* a PRF. Consider the following distinguisher D:

First, D queries $F_{A,b}(0)$ which equals b, i.e., D can compute b with just one query. As b is known, D can compute Ax by every x. A boring, but

absolutely right solution would be that D now computes A by posing queries for the unit vectors. Having A and b it is straightforward to continue.

Another solution (that does not need to query all unit vectors) would be to compute vectors v and w such that v, w and $v + w$ are pairwise distinct. As A is a linear mapping D has to check whether $Av + Aw = A(v + w)$. The winning probability for this is in fact $1 - 2^{-n}$ which is *not* negligible.

□

Q.3 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

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

Solution:

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

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

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

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

We define the working of the distinguisher D as follows, where D is given access to some oracle $\mathcal{O}$: $\{0,1\}^n \to \{0,1\}^n$ 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 $m_1, \ldots, m_{\ell_i}$, do the following

   (a) Choose a uniform initial value $ctr_i \in \{0,1\}^m$

   (b) Query $\mathcal{O}$ for $j = 1, \ldots, \ell_i$ to obtain $y_j := \mathcal{O}(ctr_i + j)$

(c) Return the ciphertext blocks $\langle ctr_i, c_1, \ldots, c_{\ell_i} \rangle := \langle ctr_i, y_1 \oplus m_1, \ldots, y_{\ell_i} \oplus m_{\ell_i} \rangle$ to A

(2) Once A outputs the messages m_0, m_1 consisting of ℓ^* blocks $m_{0,1}, \ldots, m_{0,\ell^*}$, $m_{1,1}, \ldots, m_{1,\ell^*}$, 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 $j = 1, \ldots, \ell^*$ to obtain $y_j^* := \mathcal{O}(ctr^* + j)$ Return the challenge ciphertext blocks $\langle ctr^*, c_1^*, \ldots, c_{\ell^*}^* \rangle := \langle ctr^*, y_1 \oplus m_{b,1}, \ldots, y_{\ell^*} \oplus m_{b,\ell^*} \rangle$ 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 $PrivK^{CPA}_{A,\Pi}$ or $PrivK^{CPA}_{A,\tilde{\Pi}}$, 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 $PrivK^{CPA}_{A,\Pi}(n)$ are identically distributed, and $D^{f(\cdot)}(1^n)$ and $PrivK^{CPA}_{A,\tilde{\Pi}}(n)$ are identically distributed, for uniformly chosen f. In other words,

$$\Pr_{k \leftarrow \{0,1\}^n}\left[D^{F_k(\cdot)}(1^n) = 1\right] = \Pr\left[PrivK^{CPA}_{A,\Pi}(n) = 1\right],$$
$$\Pr_{f \leftarrow Func_n}\left[D^{f(\cdot)}(1^n) = 1\right] = \Pr\left[PrivK^{CPA}_{A,\tilde{\Pi}}(n) = 1\right].$$

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

□

Q.4 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 a 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.5 The CBC construction is often used to get an encryption for larger message size. If $p : \{0,1\}^m \to \{0,1\}^m$ is a permutation, then $CBC_\ell\langle p\rangle$ is a permutation from $\{0,1\}^{\ell \cdot m}$ to $\{0,1\}^{\ell \cdot m}$ defined in the following way: for $x_1, \ldots, x_\ell \in \{0,1\}^m$, let $y_0 = 0^m$ and define $y_i = p(y_{i-1} \oplus x_i)$. Then, $CBC_\ell\langle p\rangle(x_1, \ldots, x_\ell) = (y_1, \ldots, y_\ell)$. Note that the inverse of $CBC_\ell\langle p\rangle$ can be computed in a similar way using the inverse of $p(\cdot)$.

Let $\{p_k\}$ be a pseudorandom permutation collection. Determine the CPA-security of the following two encryption schemes which are based on the CBC construction. That is, for each scheme either prove that it is CPA-secure or give an attack showing that it is not. For simplicity, we consider only the 3-block variant of the scheme (i.e., $\ell = 3$).

1. (*Padding in the end*) Given $p_k : \{0,1\}^m \to \{0,1\}^m$ and a message $x = x_1, x_2 \in \{0,1\}^{2m}$, E_k chooses $r \leftarrow_R \{0,1\}^m$ and outputs $CBC_3\langle p_k\rangle(x_1, x_2, r)$. Decrypting done in the obvious way.

2. (*Padding in the start*) Given $p_k : \{0,1\}^m \to \{0,1\}^m$ and a message $x = x_1, x_2 \in \{0,1\}^{2m}$, E_k chooses $r \leftarrow_R \{0,1\}^m$ and outputs $CBC_3\langle p_k\rangle(r, x_1, x_2)$. Decrypting done in the obvious way.

Solution:

1. This is not CPA-secure. Consider the following game with an efficient attacker A:

   1) The key k is chosen at random and fixed.
   2) A gets two different messages $m_0, m_1 \in \{0,1\}^{2m}$. Then A interacts with the encryption oracle E_k and obtains two ciphertexts y_0, y_1.
   3) A sends m_0, m_1 to the challenger and gets $c^* = E_k(m_b)$ for $b \leftarrow_R \{0,1\}$. A intercepts the first $2m$ bits of c^* as c'.
   4) If c' is identical to the first $2m$ bits of y_0, A outputs $b = 0$; otherwise, A outputs $b = 1$.

   Thus, A wins the game with probability 1.

2. This is CPA-secure (pls refer to the proof of Theorem 5.1, considering $y_1 = p_k(0^n \oplus r)$ as a random function).

□

Q.6 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

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

(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 $m = m_1 \dots m_\ell$, where $m_i \in \{0,1\}^{\frac{n}{2}}$, choose uniform $r \leftarrow \{0,1\}$, 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 $t = Mac_k(0^n 1^n) = F_k(0^n) \oplus F_k(1^n)$. Now A outputs $(1^n 0^n, t)$. This is a valid message-tag pair as $Mac_k(1^n 0^n) = F_k(1^n) \oplus F_k(0^n) = F_k(0^n) \oplus F_k(1^n) = t$, 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

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

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.7 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.8 Let (E, D) be a CPA secure scheme with key-size = message-size = n, and let $\{f_k\}$ be a collection of PRFs such that for every $k \in \{0,1\}^n$, $f_k : \{0,1\}^n \to \{0,1\}^n$. Consider the following scheme (E', D'):
Key k, k' each chosen uniformly and independently from $\{0,1\}^n$.
Encrypt $E'_{k,k'}(x) = (y, t)$ where $y = E_k(x)$ and $t = f_{k'}(y)$.
Decrypt $D'_{k,k'}(y, t) = \perp$ if $f_{k'}(y) \neq t$ and $D_k(y)$ otherwise.

1. Prove that the scheme (E', D') is CCA secure.

2. Let (E'', D'') be the same scheme except that we reuse the key for the PRFs and encryption. That is, we set $k = k'$ to be the same string chosen at random in $\{0,1\}^n$. Prove that this scheme is *not* necessarily even CPA secure! That is, show that there exists a CPA secure (E, D) and a PRF collection $\{f_k\}$ such that if we build (E'', D'') using these components then the resulting scheme is not CPA secure. Also, show that it is not CCA secure. (This example show that "reusing" or "recycling" keys in cryptography is very dangerous.)

Solution:

1. This is in fact to use the EtA construction. Suppose that A' is a T-time algorithm attacking the encryption scheme (E', D'). We will convert A' to an algorithm A that breaks the CPA security of the scheme (E, D), which will lead to a contradiction.

   The algorithm A gets black-box access to $E_k(\cdot)$ but not to $D_k(\cdot)$. The algorithm A will do the following:

   - Choose $k \leftarrow_R \{0,1\}^n$
   - Run A'
   - Whenever A' asks for an encryption of x, pass the request to the encryption box E_k to obtain $y = E_k(x)$, compute $t = f_{k'}(y)$ and give $\langle y, t\rangle$ to A. Also record this query in a table.
   - If A' asks for a decryption of $\langle y, t\rangle$ which was previously returned to it as an encryption of x then return x to A
   - If A' asks for a decryption of $\langle y, t\rangle$ which was *not* previously returned to the encryption oracle, then check if $f_{k'}(y) = t$. If check fails then return $\perp$ to A. If check succeeds then abort the computation. In this case we say that A failed to simulate A'
   - When A' sends the challenge x_1, x_2, pass it on to the sender to obtain $y = E_k(x_i)$ and give $\langle y, t\rangle$ to A', where $t = f_{k'}(y)$
   - When A' outputs a guess j, output the same guess j.

   We see that the only case that A fails to simulate the CCA attack of A' is when A' manages to produce pair $\langle y, t\rangle$ such that

   (i) $\langle y, t\rangle$ was *not* obtained as a previous response to a query x of the encryption oracle.

   (ii) $\langle y, t\rangle$ is *not* the encryption of the challenge.

   (iii) $f_{k'}(y) = t$.

   However, if A' does that then he breaks the MAC from the PRF $f_{k'}$. Indeed, because of the unique signatures property of the MAC, Properties (i) and (ii) imply that y was not previously signed by the MAC, and hence it should not be possible for A' to find a t such that $f_{k'}(y) = t$ (i.e., $Ver_{k'}(y, t) = 1$).

2. The new scheme (E'', D'') is the following:

   Key k chosen uniformly and independently from $\{0,1\}^n$.
   Encrypt $E''_k(x) = (y, t)$ where $y = E_k(x)$ and $t = f_k(y)$.
   Decrypt $D''_k(y, t) = \perp$ if $f_k(y) \neq t$ and $D_k(y)$ otherwise. Let (E, D) be the CPA-secure encryption in Theorem 5.3, i.e., $E_k(x) = (r, f_k(r) \oplus x)$ and $D_k(r, z) = f_k(r) \oplus z$. Suppose that the attacker A chooses two messages x_0, x_1 with $x_0 = 0^n$, then the ciphertext for x_0 is always $(r, f_k(r), f_k(r))$. So the attacker A can easily distinguish this case, outputs $b = 0$, and outputs $b = 1$ otherwise. Therefore, the encryption scheme (E'', D'') is *not* CPA secure. It is also not CCA secure.

□

Q.9 Prove the following modifications of basic CBC-MAC do not yield a secure MAC (even for fixed-length messages).

(1) Mac outputs all blocks $t_1, \ldots, t_\ell$, 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, $m_1, \ldots, m_\ell$, 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.10 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.11 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.

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

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

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

$$\Pr[Succ_{\tilde{A}}(n)] = \Pr[Hash - col_{\tilde{A},\tilde{\Pi}}(n) = 1] > \frac{1}{q(n)}.$$

Furthermore the case analysis above shows that

$$\Pr[Hash - col_{A,\Pi}(n) = 1 | Suss_{\tilde{A}}(n)] = 1.$$

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

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

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

□