Chapter 1

1.1. abcdefghijklmnopqrstuvwxyz $=\gg hcjkfeyvbuxzplomtgwqiasdrn$
Decryption the ciphertext provided :
CRYPTOGRAPHICSYSTEMSAREEXTREMELYDIFFICULTTOBUILDNEVERTHELESSF ORSOMEREA-
SONMANYNONEXPERTSINSISTONDESIGNINGNEWENCRYPTIONSCHEM ESTHATSEEMTOTHEM-
TOBEMORESECURETHANANYOTHERSCHEMEONEARTHTHEUNF ORTUNATETRUTHHOWEVERISTHAT-
SUCHSCHEMESAREUSUALLYTRIVIALTOBREAK

1.2. Denoting $\{0, \ldots, 25\}$ by $\mathbb{Z}_{26}$, we have:

- Gen: outputs a uniform key k from the set of bijections $p\colon \mathbb{Z}_{26} \to \mathbb{Z}_{26}$, where we are associating each letter of the English alphabet, in order, with the correspoding number in $\mathbb{Z}_{26}$.
- Enc: The encryption of the message $m = m_1 \cdots m_\ell$, where $m_i \in \mathbb{Z}_{26}$ with key k is given by:

  $$\mathsf{Enc}_k(m_1 \cdots m_\ell) = c_1 \cdots c_\ell$$

  where $c_i = k(m_i)$, i.e. we apply bijection k to m_i.
- Dec: The decryption of the ciphertext $c = c_1 \cdots c_\ell$, where $c_i \in \mathbb{Z}_{26}$ with key k is given by:

  $$\mathsf{Dec}_k(c_1 \cdots c_\ell) = k^{-1}(c_1) \cdots k^{-1}(c_\ell) = m_1 \cdots m_\ell$$

  where k^{-1} is the inverse of the function k, which exists because k is bijective.

1.3. TODO

1.4. TODO

1.5. Shift cipher: the encryption of a single character sufficies, since $c = m + k \mod 26$, so $k = c - m \mod 26$.
Monoalphabetic substitution: if the alphabet contains n characters, at least $n - 1$ distinct characters are necessary to recover the key (as the n^{th}) character is determined once the other $n - 1$ characters are). By choosing $m = m_1 \cdots m_{n-1}$ with $m_1 \neq m_2 \neq \cdots \neq m_{n-1}$, we have $c = c_1 \cdots c_{n-1} = k(m_1) \cdots k(m_{n-1})$, and we may find bijection k.
Vigenère: given a key k of length n, n characters suffice to recover the key, as each part of it can be recovered as in the shift cipher.

1.6. Since the distance that each character is shifted by is fixed, the attacker can choose `abcd` if the ciphertext contains consecutive characters (e.g. `mnop`) and `bedg` otherwise.

1.7. It is not possible with period 2. With period 3:

0	1	2	3
a	b	c	d
k_1	k_2	k_3	k_4

1	4	3	6
b	e	d	g
k_1	k_2	k_3	k_4

As for Vigenère with period 4, $|\mathcal{K}| = |\mathcal{M}|$, hence we have perfect secrecy.

1.8. TODO

Chapter 2

2.1. TODO

2.2. Enc takes a message $m \in \mathcal{M}$ and a key $k \in \mathcal{K}$, and is randomised (it gets a number of bits from some random tape that it uses as input as well). Instead of implicitly getting the random bits, we make it explicit passing them as input, by redifining the key space to $\mathcal{K} \times \mathcal{R}$ (where $\mathcal{R}$ is the set of all possible random tapes of the aximal length we could need):
Thus, Enc becomes deterministic, as it has all the randomness it needs in the new-style key.

	0	1
$(0,0,0)$	$(0,0)$	$(1,1)$
$(0,0,1)$	$(0,0)$	$(1,1)$
$(0,1,0)$	$(0,0)$	$(1,1)$
$(0,1,1)$	$(0,1)$	$(1,0)$
$(1,0,0)$	$(1,1)$	$(0,0)$
$(1,0,1)$	$(1,1)$	$(0,0)$
$(1,1,0)$	$(1,1)$	$(0,0)$
$(1,1,1)$	$(1,0)$	$(0,1)$

2.3. Consider a scheme with 1 bit of plaintext, 3 bits of key, and 2 bits of ciphertext. The two bits of ciphertext, c_0 and c_1, are obtained as follows:

$$c_0 = m_0 \oplus k_0$$
$$c_1 = (k_2 \ \wedge \ k_1) \oplus m_0 \oplus k_0$$

The possible ciphertexts can be seen in the following table:
The scheme is perfectly secure:

$$P[M=0|C=(0,0)] = \frac{P[M=0]\cdot(P[K=(0,0,0)]+P[K=(0,0,1)]+P[K=(0,1,0)])}{P[C=(0,0)]}$$
$$= \frac{P[M=0]\cdot\frac{3}{8}}{\frac{6}{16}} = P[M=0]$$

$$P[M=0|C=(0,1)] = \frac{P[M=0]\cdot P[K=(0,1,1)]}{P[C=(0,1)]}$$
$$= \frac{P[M=0]\cdot\frac{1}{8}}{\frac{2}{16}} = P[M=0]$$
$$\vdots$$
$$etc.$$

but $\frac{3}{8} = P[C=(0,0)] \neq P[C=(0,1)] = \frac{1}{8}$.

2.4. Assume that the encryption scheme is perfectly secret, and fix messages $m_0, m_1 \in \mathcal{M}$ and a ciphertext $c \in \mathcal{C}$. By Lemma 2.2 of the 1st edition of the book, we have:

$$P[C=c|M=m_0] = P[C=c] = P[C=c|M=m_1]$$

Completing the proof of the "only if" ($\Rightarrow$) direction.

Note that $P[\mathsf{Enc}_k(m)=c] = P[C=c|M=m]$, as explained in page 30. It is also worth pointing that Lemma 2.2 is an equivalent formulation of perfect secrecy, stating:
An encryption scheme $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ over a message space $\mathcal{M}$ is perfectly secret iff for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$:

$$P[C=c|M=m] = P[C=c]$$

2.5. We need to prove that an encryption scheme Π is perfectly secret iff it is perfectly indistinguishable.

($\Rightarrow$) : In what follows, we make the assumption that the adversary is deterministic. Suppose Π is perfectly secret. We need to show that $P[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1] = \frac{1}{2}$.

$$\begin{aligned}
P[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1] &= P[b' = b] \\
&= P[b' = 0 | M = m_0] \cdot P[M = m_0] + P[b' = 1 | M = m_1] \cdot P[M = m_1] \\
&= \frac{1}{2} \left(P[b' = 0 | M = m_0] + P[b' = 1 | M = m_1] \right)
\end{aligned}$$

Note that in the last step we used that $P[M = m_0] = P[M = m_1] = \frac{1}{2}$. Essentially what the adversary does is try to partition the ciphertext space $\mathcal{C}$ into two subsets $\mathcal{C}_0, \mathcal{C}_1$ such that $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1$ and $\mathcal{C}_0 \cap \mathcal{C}_1 = \emptyset$. If the attacker gets $c \in \mathcal{C}_0$ it outputs 0, else if $c \in \mathcal{C}_1$, it outputs 1. We thus proceed:

$$\begin{aligned}
&\frac{1}{2} \left(P[b' = 0 | M = m_0] + P[b' = 1 | M = m_1] \right) \\
=&\frac{1}{2} \left(\sum_{c \in \mathcal{C}_0} P[C = c | M = m_0] + \sum_{c \in \mathcal{C}_1} P[C = c | M = m_1] \right) \\
=&\frac{1}{2} \left(\sum_{c \in \mathcal{C}_0} P[C = c] + \sum_{c \in \mathcal{C}_1} P[C = c] \right) \\
=&\frac{1}{2} \left(P[c \in \mathcal{C}_0] + P[c \in \mathcal{C}_1] \right) = \frac{1}{2}
\end{aligned}$$

Note that the second and third lines are equal by an equivalent formulation of perfect secrecy, and the last equality holds since $P[c \in \mathcal{C}_0] + P[c \in \mathcal{C}_1] = 1$, because $\mathcal{C}_0$ and $\mathcal{C}_1$ are mutually exclusive and exhaustive.

($\Leftarrow$) : We prove the contrapositive, i.e. $\neg$Perfect secrecy $\Rightarrow$ $\neg$Adversarial indistinguishability. Suppose Π is not perfectly secret, then $\exists m_0', m_1' \in \mathcal{M}$ and $c' \in \mathcal{C}$ such that:

$$P[C = c' | M = m_0'] \neq P[C = c' | M = m_1']$$

(Using an equivalent formulation to the original perfect secrecy). Let $\mathcal{A}$ be an adversary that chooses m_0' and m_1'. If it receives c', $\mathcal{A}$ outputs $b' = 0$, otherwise $b' \leftarrow \{0, 1\}$ (the randomness is to ensure we can separate out the case when $C = c'$).

$$\begin{aligned}
P[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1] &= P[b = b'] \\
&= P[b = b' | M = m_0'] P[M = m_0'] + P[b = b' | M = m_1'] P[M = m_1'] \\
&= \frac{1}{2} \left(P[b = b' | M = m_0'] + P[b = b' | M = m_1'] \right) \\
P[b = b' | M = m_0'] &= P[C = c' | M = m_0'] \cdot P[b = b' | M = m_0', C = c'] \\
&\quad + P[C \neq c' | M = m_0'] \cdot P[b = b' | M = m_0', C \neq c'] \\
&= P[C = c' | M = m_0'] \cdot 1 + P[C \neq c' | M = m_0'] \cdot \frac{1}{2} \\
P[b = b' | M = m_1'] &= P[C = c' | M = m_1'] \cdot P[b = b' | M = m_1', C = c'] \\
&\quad + P[C \neq c' | M = m_1'] \cdot P[b = b' | M = m_1', C \neq c'] \\
&= P[C = c' | M = m_1'] \cdot 0 + P[C \neq c' | M = m_1'] \cdot \frac{1}{2} \\
&= P[C \neq c' | M = m_1'] \cdot \frac{1}{2}
\end{aligned}$$

Substituting back:

$$\begin{aligned}
P[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1] &= \frac{1}{2}\left(P[C = c'|M = m'_0] + P[C \neq c'|M = m'_0]\frac{1}{2} + P[C \neq c'|M = m'_1]\frac{1}{2}\right) \\
&= \frac{1}{2}P[C = c'|M = m'_0] + \frac{1}{4}\left(1 - P[C = c'|M = m'_0]\right) + \frac{1}{4}P[C \neq c'|M = m'_1] \\
&= \frac{1}{4} + \frac{1}{4}\left(P[C = c'|M = m'_0] + P[C \neq c'|M = m'_1]\right) \\
&\neq \frac{1}{4} + \frac{1}{4}\left(P[C = c'|M = m'_1] + P[C \neq c'|M = m'_1]\right) \\
&= \frac{1}{2}
\end{aligned}$$

The inequality comes from supposing that there is no perfect secrecy. Therefore:

$$P[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1] \neq \frac{1}{2}$$

Hence Π does not have adversarial indistinguishability.

2.6.

a. Take $m = 0$ and $c = 0$ as a counterexample. Then we have:

$$\begin{aligned}
P[C = c|M = m] &= P[\mathsf{Enc}_k(m) = c] \\
&= P[m + K \equiv c \pmod 5] \\
&= P[K \equiv c - m \pmod 5] \\
&= \frac{2}{6} = \frac{1}{3}
\end{aligned}$$

k (key) \ m (message)	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3
5	0	1	2	3	4

But $P[C = c] = \frac{6}{36} = \frac{1}{6}$. Thus, we found a pair $m \in \mathcal{M}$ and $c \in \mathcal{C}$ for which $P[C = c|M = m] \neq P[C = c]$, so the scheme is not perfectly secret.

b. TODO

2.7. By the contrapositive of theorem 2.10, if $|\mathcal{K}| < |\mathcal{M}|$ then the encryption scheme is not perfectly secret. Alternatively, we can see that for any $m \in \mathcal{M}$ and $c \in \mathcal{C}$ such that $m = c$, $P[C = c|M = m] = 0$ (since we are missing precisely the key that makes $c = m \oplus k = m$). However, $P[C = c] \neq 0$. The intuition for why key 0^ℓ is no different from other keys is that $c = 0^\ell \oplus m = m$ is equivalent to $c = k' \oplus m'$, for any $k' = c \oplus m'$, and there is no reason why an adversary should assume that the key is 0^ℓ instead of k'.

2.8.

a.

$$\begin{aligned}
P[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1] &= P[b = b'] \\
&= P[b = 0] \cdot P[\mathcal{A} \text{ outputs } 0|b = 0] + P[b = 1] \cdot P[\mathcal{A} \text{ outputs } 1|b = 1] \\
&= \frac{1}{2}\left(P[\mathcal{A} \text{ outputs } 0|b = 0] + P[\mathcal{A} \text{ outputs } 1|b = 1]\right)
\end{aligned}$$

Then, we have:

$$P[\mathcal{A} \text{ outputs } 0|b = 0] = \frac{1}{3} \cdot \underbrace{\frac{26}{26}}_{(a)} + \frac{1}{3} \cdot \underbrace{\frac{26}{26^2}}_{(b)} + \frac{1}{3} \cdot \underbrace{\frac{26^2}{26^3}}_{(c)} = \frac{14}{39}$$

where:

a. when $|k| = 1$, $\mathcal{A}$ always wins.

b. when $|k| = 2$, $\mathcal{A}$ wins when both symbols of the key equal each other ($k_1 = k_2$). This happens in $\frac{26}{26^2}$ keys.

c. when $|k| = 3$, $\mathcal{A}$ wins when two symbols of the key equal each other. This happens in $\frac{26^2}{26^3}$ keys.

Also:

$$P[\mathcal{A} \text{ outputs } 1|b = 1] = \frac{1}{3} \cdot \underbrace{\frac{26}{26}}_{(a)} + \frac{1}{3} \cdot \underbrace{\frac{26 \cdot 25}{26^2}}_{(b)} + \frac{1}{3} \cdot \underbrace{\frac{26^2 \cdot 25}{26^3}}_{(c)} = \frac{38}{39}$$

where:

a. when $|k| = 1$, $\mathcal{A}$ always wins.

b. when $|k| = 2$, $\mathcal{A}$ loses when $k_1 = k_2 + 1$ (since $\mathsf{Enc}_k(abb) = c_1 c_{1\,c3}$). This happens in 26 cases, so we care about the remaining $26^2 - 26 = 26 \cdot 25$ of the 26^2 cases.

c. when $|k| = 3$, $\mathcal{A}$ loses oncd more when the symbols of k are consecutive, leading to $26^3 - 26^2 = 26^2 \cdot 25$ of the 26^3 cases.

Thus, substituting in our derivation:

$$\frac{1}{2}\left(\frac{14}{39} + \frac{38}{39}\right) = \frac{2}{3}$$

b. TODO

2.9.

a. The easiest proof is by Shannon's theorem: we have $|\mathcal{C}| = |\mathcal{M}| = |\mathcal{K}| = 26$, each key in $\mathbb{Z}_{26}$ is chosen with equal probability ($\frac{1}{26}$), and for every $m \in \mathbb{Z}_{26}$ and $c \in \mathbb{Z}_{26}$ there is a unique key k such that $\mathsf{Enc}_k(m) = c$, namely $k = c - m \pmod 2 6$. Alternatively, we can show that for each m, c:

$$P[C = c|M = m] = P[\mathsf{Enc}_k(m) = c] = P[K = c - m \pmod 2 6] = \frac{1}{26}$$

Thus $\forall m_0, m_1 \in \mathcal{M}$ and $\forall c \in \mathcal{C}$, $P[C = c|M = m_0] = P[C = c|M = m_1]$, and we have perfect secrecy.

b. By the limitation of perfect secrecy, $|\mathcal{M}| \leq |\mathcal{K}| = n!$, where n is the number of symbols in the alphabet ($n = 26$ for English); the factorial is because that's the number of permutations on an n-element set (in particular, $\mathbb{Z}_{26}$). So we have an upper bound for $|\mathcal{M}|$. Now take the set $\mathcal{M}$ of all strings of 26 characters without repeating any. Clearly $|\mathcal{M}| = 26!$. Once more, we use Shannon's theorem:

1. $|\mathcal{K}| = |\mathcal{M}|$, but also $|\mathcal{M}| = |\mathcal{C}|$ since any permutation on characters will map a string of 26 nonrepeated letters to another.
2. Every key is chosen with equal probability, namely $\frac{1}{26!}$.
3. For every $m \in \mathcal{M}$ and $c \in \mathcal{C}$, $\exists! k \in Ks$ such that $\mathsf{Enc}_k(m) = c$, since m and c define a unique permutation on all the letters of the alphabet.

Therefore, the largest message space $\mathcal{M}$ for which the monoalphabetic cipher provides perfect secrecy is $n!$, for an n-element set.

c. For an n-element alphabet, the Vigenère cipher using (fixed) period t has $|\mathcal{K}| = n^t$. If we encrypt messages of length t, then $|\mathcal{M}| = n^t$ too. Clearly, we also have $|\mathcal{C}| = n^t$.

Again, by Shannon's theorem we see that every key is chosen with equal probability ($\frac{1}{n^t}$), and for each pair of plaintext and ciphertext, there is a unique key such that $\mathsf{Enc}_k(m) = c$.

2.10. A simple way is the following: Let Π be a scheme satisfying definition 2.5. Then by Lemma 2.6 Π is perfectly secret, so by theorem 2.10, $|\mathcal{K}| \geqslant |\mathcal{M}|$. As for an $\mathcal{A}$ for which $P[\mathrm{PrivK}^{\mathrm{eav}}_{\mathcal{A},\Pi} = 1] > \frac{1}{2}$, let Π be an arbitrary encryption scheme with $|\mathcal{K}| < |\mathcal{M}|$.
TODO finish

2.11. Let $\mathcal{M} = \{0,1\}^n$ and $\mathcal{K} = \{0,1\}^{n-t}$, with $\mathsf{Enc}_k(m) = [m]_{1,n-t} \oplus k$, i.e. xor the first $(n-t)$ bits of m with the key k. $\mathsf{Dec}_k(c) = (c||0^t)(k||r)$, where r is a random pad of t bits. Since we have $\frac{1}{2^t}$ chances that r is precisely the missing part of the message, $P[\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m] = \frac{1}{2^t}$, so $P[\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m] \geqslant \frac{1}{2^t}$. The perfect secrecy of this scheme follows from the proof of the one-time pad (this is exactly a one-time pad on the first $(n-t)$ bits of the message). Lower bound: $2^{n-t} = |\mathcal{M}| \cdot 2^{-t} \leqslant |\mathcal{K}|$.

Chapter 3

3.1.

1. Let p be a positive polynomial. Since $2p$ is also a positive polynomial and negl_1 and negl_2 are negligible:

$$\exists N_1, N_2 \left(\forall n \geqslant N_1 \left(\mathrm{negl}_1(n) < \frac{1}{2p(n)} \right) \wedge \forall n \geqslant N_2 \left(\mathrm{negl}_2(n) < \frac{1}{2p(n)} \right) \right)$$

   Choose $N_3 = \max(N_1, N_2)$, then $\forall n \geqslant N_3$ we have:

$$\mathrm{negl}_3(n) = \mathrm{negl}_1(n) + \mathrm{negl}_2(n) < \frac{1}{2p(n)} + \frac{1}{2p(n)} = \frac{1}{p(n)}$$

2. Let p, q be two positive polynomials. Since $p \cdot q$ is also a positive polynomial and negl_1 is negligible:

$$\exists N_1 \left(\forall n \geqslant N_1 \left(\mathrm{negl}_1 < \frac{1}{q(n)p(n)} \right) \right)$$

   Then $\forall n \geqslant N_1 \left(\mathrm{negl}_4 = p(n) \cdot \mathrm{negl}_1(n) < \frac{1}{q(n)} \right)$.

3.2. Let $q(n)$ be a polynomial such that for any $k \leftarrow \mathsf{Gen}(1^n)$, $|\mathsf{Enc}_k(0)| \leqslant q(n)$. Such a polynomial exists because the encryption algorithm must run in an amount of time polynomial in n. Since the maximum encrypted length of 0 is bounded by $q(n)$, we would like our adversary to choose $m_0 = 0$, and m_1 so that m_1 will always encrypt to a string of length greater than $q(n)$. If the adversay can to this, it becomes trivial to determine which message was encrypted, i.e. $P[\mathrm{PrivK}^{\mathrm{eav}}_{\mathcal{A},\Pi} = 1] = 1$, thus the definition cannot be satisfied. Consider all strings of length $q(n)+2$. Since there are $2^{q(n)+2}$ such strings, and fewer than $2^{q(n)+1}$ strings of length $\leqslant q(n)$, there must be some string $s \in \{0,1\}^{q(n)+2}$ that can only encrypt to strings of length $> q(n)$. If the adversary chooses $m_1 = s$, then he can always win the indistinguishability experiment, so Π cannot satisfy the definition, as desired.

3.3. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a scheme that is secure with respect to the original defintion 3.8 (for messages of equal length). Construct a scheme $\Pi' = \left(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}'\right)$ such that:

- Given m, with $|m| \leqslant \ell(n)$, then:

$$\mathsf{Enc}'(m) = \begin{cases} \mathsf{Enc}(0^{\ell - |m| - 1}1||m), & \text{if } |m| < \ell(n) \\ \mathsf{Enc}(m), & \text{if } |m| = \ell(n) \end{cases}$$

- Dec' applies Dec to the ciphertext, and parses the result as $0^t1||m$ for $t \geqslant 0$. It outputs m.

A complete answer to this exercise requires a proof showing that the existence of an adversary breaking Π' with respect to the modified definitions implies the existence of an adversary breaking Π with respect to definition 3.8.
Informally: Given an adversary $\mathcal{A}'$ who breaks Π', we construct an adversary $\mathcal{A}$ who takes the pair of plaintexts m_0, m_1 output by $\mathcal{A}'$ and pads them in the same way as Enc' would. Then it outputs the padded messages to be encrypted. Observe that $\mathcal{A}$ outputs equal length messages, as required. Furthermore, if $\mathcal{A}'$ can correctly guess b with probability greater than $\frac{1}{2}$, then this guess will also be correct for $\mathcal{A}$ with the same probability.

3.4. Assume the scheme has indistinguishabile encryption in the presence of an eavesdropper (def 3.8), i.e.:

$$P[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1] \leqslant \frac{1}{2} + (n)$$

TODO finish

Chapter 4

4.1. TODO finish

4.2. TODO finish

4.3. TODO finish

4.4. TODO finish

4.5. TODO finish

4.6. TODO finish

4.7. Let F be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. In each case Gen outputs a uniform $k \in \{0,1\}^n$. Let ⟨i⟩denote an n/2-bit encoding of the integer i.

1. To authenticate a message

$$m = m_1, \ldots, m_l$$

where

$$m_i \in \{0,1\}^n$$

, compute

$$t := F_k(m_1) \oplus \cdots \oplus F_k(m_l)$$

.

2. To authenticate a message

$$m = m_1, \ldots, m_l$$

where

$$m_i \in \{0,1\}^{n/2}$$

, compute

$$t := F_k(\langle 1\rangle || m_1) \oplus \cdots \oplus F_k(\langle l\rangle || m_l)$$

.

3. To authenticate a message

$$m = m_1, \ldots, m_l$$

where

$$m_i \in \{0,1\}^{n/2}$$

, choose uniform

$$r \leftarrow \{0,1\}^n$$

, compute

$$t := F_k(r) \oplus F_k(\langle 1\rangle || m_1) \oplus \cdots \oplus F_k(\langle l\rangle || m_l)$$

, and let the tag be

$$\langle r, t\rangle$$

4.8. Let F be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages.In each case Gen outputs a uniform $k \in \{0,1\}^n$. Let ⟨i⟩denote an n/2-bit encoding of the integer i.

1. To authenticate a message

$$m = m_1, \ldots, m_l$$

where

$$m_i \in \{0,1\}^n$$

, compute

$$t := F_k(m_1) \oplus \cdots \oplus F_k(m_l)$$

. 2. To authenticate a message

$$m = m_1, \ldots, m_l$$

where

$$m_i \in \{0,1\}^{n/2}$$

, compute

$$t := F_k(\langle 1 \rangle || m_1) \oplus \cdots \oplus F_k(\langle l \rangle || m_l)$$

. 3. To authenticate a message

$$m = m_1, \ldots, m_l$$

where

$$m_i \in \{0,1\}^{n/2}$$

, choose uniform

$$r \leftarrow \{0,1\}^n$$

, compute

$$t := F_k(r) \oplus F_k(\langle 1 \rangle || m_1) \oplus \cdots \oplus F_k(\langle l \rangle || m_l)$$

, and let the tag be

$$\langle r, t \rangle$$

Answer :
* Part 1 :
Reorder the blocks in "m" and the tag doesn't change.
* Part 2 :
Query
*

$$m^1 = m_1 || m_2$$

, tag

$$t_1 = F_k(\langle 1 \rangle || m_1) \oplus F_k(\langle 2 \rangle || m_2)$$

*

$$m^2 = m_3 || m_2$$

, tag

$$t_2 = F_k(\langle 1 \rangle || m_3) \oplus F_k(\langle 2 \rangle || m_2)$$

*

$$m^3 = m_3 || m_4$$

, tag

$$t_3 = F_k(\langle 1 \rangle || m_3) \oplus F_k(\langle 2 \rangle || m_4)$$

Thus

$$m^* = m^1 \oplus m^2 \oplus m^3 = m_1 || m_4$$

, tag

$$t = t_1 \oplus t_2 \oplus t_3 = F_k(\langle 1 \rangle || m_1) \oplus F_k(\langle 2 \rangle || m_4)$$

.
*

$$Part3:$$

Let

$$m \in \{0,1\}^{n/2}$$

. When choosing

$$r = \langle 1 \rangle || m$$

,

$$t = F_k(r) \oplus F_k(\langle 1 \rangle || m) = 0^n$$

.
Thus

$$t = \langle \langle 1 \rangle || m, 0^n \rangle$$

will be a valid tag for "m".

4.9. Let "F" be a pseudorandom function. 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

$$F_k(m_1) || F_k(F_k(m_2))$$

.
Answer:
Query
*

$$m^1 = m_1^* || m_1^*$$

,

$$t^1 = t_1^1 || t_2^1 = F_k(m_1^*) || F_k(F_k(m_1^*))$$

*

$$m^2 = m_2^* || m_2^*$$

,

$$t^2 = t_1^2 || t_2^2 = F_k(m_2^*) || F_k(F_k(m_2^*))$$

Hence for

$$m^* = m_1^* || m_2^*$$

,

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

4.10. TODO finish

4.11. TODO finish

4.12. TODO finish

4.13. TODO finish

4.14. Prove that the following modifications of basic CBC-MAC do not yield a secure MAC (even for fixed-length messages) :
1. Mac outputs all blocks

$$t_1, \ldots, t_l$$

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

$$t \in \{0,1\}^n$$

, run basic CBC-MAC over the "message"

$$t_0, m_1, \ldots, m_l$$

, and output the tag

$$\langle t_0, t_l \rangle$$

. Verification is done in the natural way.
The Answer :
* Part 1:
Query
*

$$m^1 = B_0 || B_1$$

,

$$t^1 = t_0 || t_1$$

*

$$m^2 = B_2 || B_3$$

,

$$t^2 = t_2 || t_3$$

We know

$$F_k(B_0) = t_0$$

and

$$F_k(B_2) = t_2$$

. Hence

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

Let

$$t_0 \oplus B_2^* = B_2$$

, i.e.,

$$B_2^* = t_0 \oplus B_2$$

. Then

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

Therefore,

$$\langle B_0 || t_0 \oplus B_2, t_0 || t_2 \rangle$$

is a valid pair of message and tag.
* Part 2:
Query

*

$$m^1 = B_0||B_1$$

,

$$t^1 = \langle r_1, t_1 \rangle$$

*

$$m^2 = B_2||B_3$$

,

$$t^2 = \langle r_2, t_2 \rangle$$

Hence for

$$m^* = B_0||B_1||t_2 \oplus r_2||B_2||B_3$$

,

$$t^* = \langle r, t_2 \rangle$$

should be a valid tag.

4.15. Show that appending the message length to the end of the message before applying basic CBC-MAC does not result in a secure MAC for arbitrary-length messages.
The Answer :
Query
*

$$m_1 = B_0||B_1$$

,

$$t_1 = MAC_k(m_1||\langle|m_1|\rangle)$$

*

$$m_1^* = B_0^*||B_1^*$$

,

$$t_1^* = MAC_k(m_1^*||\langle|m_1^*|\rangle)$$

*

$$|m_1^*| = |m_1|$$

*

$$m_2 = m_1||\langle|m_1|\rangle||B_2||B_3$$

,

$$t_2 = MAC(m_2||\langle|m_2|\rangle)$$

To be specific, the process of computing

$$t_2$$

for message

$$m_2$$

is listed below:
*

$$c_0 = F_k(B_0)$$

*

$$c_1 = F_k(c_0 \oplus B_1)$$

*

$$t_1 = F_k(c_1 \oplus \langle|m_1|\rangle)$$

*

$$c_3 = F_k(t_1 \oplus B_2)$$

*

$$c_4 = F_k(c_3 \oplus B_3)$$

*

$$t = F_k(c_4 \oplus \langle |m_2| \rangle)$$

Hence, if we change

$$m_1$$

to

$$m_1^*$$

,
*

$$c_0^* = F_k(B_0^*)$$

*

$$c_1^* = F_k(c_0^* \oplus B_1^*)$$

*

$$t_1^* = F_k(c_1^* \oplus \langle |m_1^*| \rangle)$$

In order to keep the result of MAC, it must hold that

$$t_1 \oplus B_2 = t_1^* \oplus B_2^*$$

. Thus

$$B_2^* = t_1 \oplus B_2 \oplus t_1^*$$

Therefore
*

$$c_3^* = F_k(t_1^* \oplus B_2^*) = F_k(t_1^* \oplus t_1 \oplus B_2 \oplus t_1^*) = F_k(t_1 \oplus B_2) = c_3$$

*

$$c_4^* = F_k(c_3^* \oplus B_3) = F_k(c_3 \oplus B_3) = c_4$$

*

$$t^* = F_k(c_4^* \oplus \langle |m_2^*| \rangle) = F_k(c_4 \oplus \langle |m_2| \rangle) = t$$

*

$$|m_2^*| = |m_2|$$

can be easily get since

$$|m_1^*| = |m_1|$$

Hence we get a message and its valid tag

$$\langle m^*, t^* \rangle$$

where

$$m^* := m_1^* || \langle |m_1^*| \rangle || t_1 \oplus B_2 \oplus t_1^* || B_3 t^* = t$$

4.16. Show two types of forgery attacks for authenticated encryption scheme CBC-XOR.
Given a pseudorandom permutation F
$Gen : k \ll 0, 1^n$
Enc: On input a message m = $B_0||B_1||...||B_l$ and a key k, uniformly generate an IV$\ll$0, 1^m
1. Compute $B_{l+1} = B_0||B_1||...||B_l$
2. Do CBC encryption on m —— B_{l+1}using k and IV
- Output ciphertext c := IV —— $c_0||c_1||...||c_l||c_{l+1}$
Dec: On input a ciphertext c = IV —— $c_0||c_1||...||c_l||c_{l+1}$and a key k
1. Do CBC decryption on $c_0||c_1||...||c_l||c_{l+1}$using k and IV

2. Check if $B_{l+1} = B_0||B_1||...||B_l$−If true, output plaintext $B_0||B_1||...||B_l$−If false, output error
Answers :
Method 1 - Truncation
Query

$$m = B_0||B_1||(B_0 \oplus B_1)$$

and obtain the ciphertext

$$c = IV||c_0||c_1||c_2||c_3$$

.
Thus

$$c^* = IV||c_0||c_1||c_2$$

should be a valid ciphertext for

$$m^* = B_0||B_1$$

Method 2 - Swap
Query

$$m = B_0||B_1||B_2$$

and obtain the ciphertext

$$c = IV||c_0||c_1||c_2||c_3$$

Thus
*

$$F_k(IV \oplus B_0) = c_0$$

*

$$F_k(c_0 \oplus B_1) = c_1$$

*

$$F_k(c_1 \oplus B_2) = c_2$$

*

$$F_k(c_2 \oplus B_0 \oplus B_1 \oplus B_2) = c_3$$

Hence

$$c^* = IV||c_1||c_0||c_2||c_3$$

should be a valid tag for

$$m^* = B_1^*||B_0^*||B_2^*$$

, where
*

$$B_0^* = c_0 \oplus B_1 \oplus IV$$

*

$$B_1^* = IV \oplus B_0 \oplus c_1$$

*

$$B_2^* = c_1 \oplus B_2 \oplus c_0$$

*

$$B_0^* \oplus B_1^* \oplus B_2^* = c_0 \oplus B_1 \oplus IV \oplus IV \oplus B_0 \oplus c_1 \oplus c_1 \oplus B_2 \oplus c_0 = B_0 \oplus B_1 \oplus B_2$$

4.17. TODO finish

Chapter 5

5.1. TODO

5.2. TODO

5.3. TODO

5.4. TODO

5.5. Problem
Let Gen,H be a collision-resistant hash function. Is Gen,$\hat{H}$ defined by

$$\hat{H}^s(x) \stackrel{def}{=} H^s(H^s(x))$$

necessarily collision resistant?
* Solution
Assuming that $\hat{H}$ is not collision-resistent, i.e.

$$\exists x \neq y, \hat{H}^s(x) = \hat{H}^s(y)$$

Thus

$$H^s(H^s(x)) = H^s(H^s(y))$$

* If

$$H^s(x) = H^s(y)$$

,

$$(x, y)$$

is a pair of collision for

$$H$$

* If

$$H^s(x) \neq H^s(y)$$

, let

$$x' = H^s(x)$$

,

$$y' = H^s(y)$$

. *

$$H^s(H^s(x)) = H^s(H^s(y))$$

,

$$(x', y')$$

is a pair of collision for

$$H$$

Therefore, $\hat{H}$ is not collision-resistent implies H is not collision-resistent. Then H is collision-resistent implies $\hat{H}$ is collision-resistent.

5.6. TODO

5.7. TODO