Introduction to cryptology
(GBIN8U16)
Final Examination

2019-05-02

Instructions

The duration of this examination is three hours. All exercises are independent, and they may be solved in any order. Answers to the questions must be detailed and complete to get maximum credit. The full scale is not determined yet: it may not be necessary to answer all questions in order to obtain a perfect mark.

Exercise 1: Hash functions

In the following questions, $\mathcal{H} : \mathcal{I} \to \{0,1\}^n$ is a cryptographic hash function, where $\mathcal{I} = \bigcup_{\ell=0}^{2^N} \{0,1\}^\ell$. We recall the two following definitions:

— A *second preimage attack* on $\mathcal{H}$ is an algorithm that on input $m \in \mathcal{I}$ returns $m' \neq m \in \mathcal{I}$ s.t. $\mathcal{H}(m') = \mathcal{H}(m)$.

— A *collision attack* on $\mathcal{H}$ is an algorithm that returns m, $m' \neq m \in \mathcal{I}$ s.t. $\mathcal{H}(m) = \mathcal{H}(m')$.

Q. 1:

1. Give an algorithm for a second preimage attack. What is its expected running time (in function of n) for a perfectly random function $\mathcal{H}$ (no justification is necessary)?
2. What is the average complexity of a collision attack for a perfectly random function $\mathcal{H}$?
3. Give the specifications of a hash function $\mathcal{H}' : \mathcal{I} \to \{0,1\}^n$ for which every pair of distinct messages forms a collision. Is it possible to efficiently find second preimages for this function?

We informally call a hash function $\mathcal{H}$ *preimage-resistant* (resp. *collision-resistant*) if there is no "efficient" (first or second) preimage attack (resp. collision attack) on $\mathcal{H}$.

Q. 2:

1. Show that an adversary having a black box access to an efficient second preimage attack can perform an efficient collision attack. Is the converse true?
2. Is it possible for a hash function to be collision-resistant but not preimage-resistant?
3. Let $\mathcal{H}$ be such that the best collision attack on it is a generic attack. What can you say about the complexity of preimage attacks on $\mathcal{H}$?

Exercise 2: Compression functions

In the following questions, $\mathcal{F} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a compression function, and $\mathcal{E} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a block cipher with n-bit keys and blocks. We give the following definitions:

— A *first preimage attack with known chaining value* on $\mathcal{F}$ is an algorithm that on input $(h,t) \in \{0,1\}^n \times \{0,1\}^n$ returns $m \in \{0,1\}^n$ s.t. $\mathcal{F}(h,m) = t$.

— A *key-recovery attack with one known plaintext* on $\mathcal{E}$ is an algorithm that on input $(m, c = \mathcal{E}(k,m)) \in \{0,1\}^n \times \{0,1\}^n$ (where k is drawn uniformly at random in $\{0,1\}^n$) returns k' s.t. $\mathcal{E}(k',m) = c$.

Q. 1:

1. Give an example of block cipher for which in the above key-recovery attacks, one always has $k' = k$. Give another example where with high probability $k' \neq k$.
2. Recall the definition of an ideal block cipher.
3. What is the average complexity of the above key-recovery attack on $\mathcal{E}$ if it is an ideal block cipher?

Q. 2: A designer proposes to build a compression function $\mathcal{F}$ from a block cipher $\mathcal{E}$ as $\mathcal{F}(h,m) := \mathcal{E}(h,m)$.

1. Give an efficient preimage attack on $\mathcal{F}$.

Q. 3: Another designer proposes to build a compression function $\mathcal{F}$ from a block cipher $\mathcal{E}$ as $\mathcal{F}(h,m) := \mathcal{E}(m,h)$.

1. Let $A \Rightarrow B$ be a logical proposition. Give its contrapositive.
2. Explain why there is no efficient preimage attack on $\mathcal{F}$ if $\mathcal{E}$ is an ideal block cipher.

Exercise 3: CBC-MAC

In the following questions, $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ is a block cipher, $\mathcal{CBC}\text{-}\mathcal{E} : \{0,1\}^\kappa \times \mathcal{I} \to \mathcal{I}$ denotes the CBC encryption *without padding* of a message $m = m_1 || \cdots || m_\ell$ of size $L = \ell n$ *with zero IV*, i.e. $\mathcal{CBC}\text{-}\mathcal{E}(k,m) = c_1 || \cdots || c_\ell$ with $c_1 = \mathcal{E}(k, m_1)$, $c_{i>1} = \mathcal{E}(k, m_i \oplus c_{i-1})$, and $\mathcal{CBC}\text{-}\mathcal{F} : \{0,1\}^\kappa \times \mathcal{I} \to \{0,1\}^n$ denotes $k, m \mapsto \textsc{LastBlock}(\mathcal{CBC}\text{-}\mathcal{E}(k,m))$, the last block of $\mathcal{CBC}\text{-}\mathcal{E}(k,m)$, i.e. $\mathcal{CBC}\text{-}\mathcal{F}(k,m) = c_\ell$ with $c_1 = \mathcal{E}(k, m_1)$, $c_{i>1} = \mathcal{E}(k, m_i \oplus c_{i-1})$.

Q. 1: In this question, we fix $\mathcal{I}$ to $\{0,1\}^{2n}$, i.e. we consider messages of exactly $2n$ bits.

1. Is $\mathcal{CBC}\text{-}\mathcal{E}$ invertible?
2. Is $\mathcal{CBC}\text{-}\mathcal{F}$ invertible?
3. Assuming that $\mathcal{E}$ is a "good" block cipher, is $\mathcal{CBC}\text{-}\mathcal{E}$ a semantically secure (IND-CPA) encryption scheme?

We recall briefly that a "good" MAC should be resistant to *existential forgery attacks*, i.e. it should be "hard" for an adversary to compute the tag of a message that was never queried before.

Q. 2: In this question, we fix $\mathcal{I}$ to $\{0,1\}^n \cup \{0,1\}^{2n}$.

1. Let $k \in \{0,1\}^\kappa$, $m \in \{0,1\}^n$, $t = \mathcal{E}(k,m)$. What is $\mathcal{CBC}\text{-}\mathcal{F}(k,m)$ equal to?
2. Let $m' = m||(m \oplus t)$. What is $\mathcal{CBC}\text{-}\mathcal{F}(k,m')$ equal to?
3. Give an existential forgery attack for $\mathcal{CBC}\text{-}\mathcal{F}$.
4. What are the time complexity, data (query) complexity, and advantage of your attack?

We now redefine $\mathcal{CBC}\text{-}\mathcal{E}$ (and $\mathcal{CBC}\text{-}\mathcal{F}$ accordingly) to accommodate messages whose length is not necessarily a multiple of n. To do so, we define a padding function Pad in the following way:

Let $m = m_1||\cdots||m_\ell||m_f$ be an L-bit message, $L = \ell n + r$, $0 \leq r < n$, where $m_1, \ldots, m_\ell$ are n-bit long and m_f is r-bit long (and possibly equal to the null string ε if L is a multiple of n); we denote by L_n the n-bit string that represents the number L (which we assume to be less than 2^n) written in base 2, and by 0^x the x-bit string made only of zeroes (if $x = 0$, then 0^x is the null string). Then $\text{Pad}(m) := m_1||\cdots||m_\ell||(m_f||1||0^{n-(r+1)})||L_n$.

We then define $\mathcal{CBC}\text{-}\mathcal{E} : \{0,1\}^\kappa \times \mathcal{I} \to \mathcal{I}$ for a message m as the CBC encryption of $\text{Pad}(m)$ with zero IV for messages of length multiple of n.

Q. 3:

1. Let $L = kn$ (i.e. $r = 0$); what is the length of $\text{Pad}(m)$?
2. What is the maximum length of $\text{Pad}(m)$ in function of L?
3. Is the modified version of $\mathcal{CBC}\text{-}\mathcal{E}$ invertible?

Q. 4:

1. Does your attack of **Q. 2** apply without any change to $\mathcal{CBC}\text{-}\mathcal{F}$ when the modified $\mathcal{CBC}\text{-}\mathcal{E}$ is used?

Q. 5: Let m, $m' \neq m$ be two messages of equal length, $t = \mathcal{CBC}\text{-}\mathcal{F}(m)$, $t' = \mathcal{CBC}\text{-}\mathcal{F}(m')$, $a = \text{Pad}(m)||0^n||s$, $t'' = \mathcal{CBC}\text{-}\mathcal{F}(a)$, $a' = \text{Pad}(m')||(t \oplus t')||s$.

1. Give an existential forgery attack for $\mathcal{CBC}\text{-}\mathcal{F}$ that from m, m', t, t', a, t'' allows to predict the tag for a'.

Exercise 4: RSA

In the following questions, $N = pq$ with p, q two distinct primes; $e, d \in (\mathbb{Z}/\varphi(N)\mathbb{Z})^\times \backslash \{1\}$ such that $ed \equiv 1 \mod \varphi(N)$; the RSA permutation RSA-P : $\mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z}$ is defined by $m \mapsto m^e \mod N$, and its inverse RSA-P^{-1} is defined by $m \mapsto m^d \mod N$.

Q. 1:

1. Give the definition of $\varphi(N)$ and its value in function of p and q.
2. Explain how the knowledge of p, q and e allows to compute d efficiently.

Q. 2: In the following questions, we take the public value of e to be equal to 3, and $N \approx 2^{3072}$.

1. Why is it easy to compute m from RSA-P(m) when $0 \leq m < 2^{1000}$?
2. How does one solve this issue in practice (i.e. how does one convert RSA-P to a semantically secure encryption scheme)?

Q. 3:

1. Give an efficient algorithm to factor N when the difference between p and q is very small. (Hint: consider the following small example when $N \approx 2^{1024}$:
   $p =$ 13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084171
   $q =$ 13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084241
   $\sqrt{N} =$ 13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084205.)

Exercise 5: Discrete logarithms

In the following questions, $\mathbb{G}$ is a finite cyclic group of prime order p (meaning that it contains p elements), g denotes one of its generators, and $h \neq g$ another element of $\mathbb{G}$.

Q. 1:

1. Give an example of a finite cyclic group, and specify its order and whether it is prime.
2. Under which condition is h a generator of $\mathbb{G}$?
3. Give the definition of the discrete logarithm of h with respect to g.
4. Is the map $[\![0, p-1]\!] \to \mathbb{G}$, $x \mapsto g^x$ injective? What if we take $x \in [\![0, p]\!]$ instead?
5. Give an algorithm that computes the inverse of an element in $\mathbb{G}$.
6. Is the map $\mathbb{G} \to \mathbb{G}$, $x \mapsto gx$ a permutation for all g? If not, under which condition on g is it one?

Q. 2: We define the *decisional Diffie-Hellman problem* (DDH) as follows: an adversary is given one of the two triples $(g^a,\ g^b,\ g^{ab})$, with $a, b \xleftarrow{\$} [\![0, p-1]\!]$ or $(g^a,\ g^b,\ g^c)$, with $a, b, c \xleftarrow{\$} [\![0, p-1]\!]$, each with probability 0.5. The adversary wins if it correctly guesses which triple it was given.

1. Show that an adversary can solve DDH with advantage close to one by computing a small number of discrete logarithms (you don't need to precisely compute the advantage).
2. Is DDH a hard problem for the group $\mathbb{G}$ if $p \approx 2^{128}$?

Q. 3: We define the *classic textbook Elgamal* public encryption scheme as follows: a *receiver* picks a private key $a \xleftarrow{\$} [\![0, p-1]\!]$, computes $k = g^a$ and publishes it as a public key. A *sender* wishing to send a message $m \in \mathbb{G}$ to the receiver picks $b \xleftarrow{\$} [\![0, p-1]\!]$ and sends (g^b, mk^b) to the receiver.

1. Explain how the receiver can decrypt a message sent by the sender.
2. Show that if DDH is not hard in $\mathbb{G}$, then there is an efficient adversary that can attack this encryption scheme with respect to the IND-CPA security notion.
3. Is the following scenario possible: 1) computing discrete logarithms in $\mathbb{G}$ is "hard"; 2) classic textbook Elgamal with the group $\mathbb{G}$ is "not IND-CPA"?