Introduction to cryptology (GBIN8U16) Final Examination

2022-05-06

Instructions

- One two-sided A4 page of (handwritten or typed) notes allowed.
- *Except indicated otherwise, answers must be carefully justified to get maximum credit.*
- Not all questions are independent, but you may admit a result from a previous question by clearly stating it.
- You may answer in English or French.
- Duration: 3 hours.

Notation & definitions

We recall some notation and definitions. The definitions are provided for context for Exercises 1 and 4. However, since no formal proofs need to be given in those exercises, most of the details given here can actually be ignored.

- For any finite set $\mathcal{S}$, we write $X \twoheadleftarrow \mathcal{S}$ to mean that the random variable X is sampled uniformly from $\mathcal{S}$. Furthermore, in notation such as $X \twoheadleftarrow \mathcal{S}, Y \twoheadleftarrow \mathcal{S}$, the samplings of X and Y are independent (except specified otherwise).
- $\cdot||\cdot$ denotes string concatenation.

Definition 1 (PRP advantage)**.** Let $E : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$ be a block cipher over the finite set $\mathcal{X}$. The *PRP advantage of* E is defined as: $\mathrm{Adv}_E^{\mathrm{PRP}}(q,t) =$

$$\begin{aligned}\max_{A_{q,t}} | &\Pr[A_{q,t}^{\mathbb{O}}() = 1 : \mathbb{O} \twoheadleftarrow \mathrm{Perms}(\mathcal{X})] \\ &- \Pr[A_{q,t}^{\mathbb{O}}() = 1 : \mathbb{O} = E(k,\cdot), k \twoheadleftarrow \mathcal{K}]|\end{aligned}$$

Where $\mathrm{Perms}(\mathcal{X})$ denotes the set of all permutations over the finite set $\mathcal{X}$, and $A_{q,t}^{\mathbb{O}}$ denotes an algorithm that runs in time t and makes q queries to the oracle $\mathbb{O}$ it is given access to.

Definition 2 (IND-CPA (PKC))**.** Let $(\mathrm{Enc}, \mathrm{Dec})$ be a public-key encryption scheme (where $\mathrm{Enc} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}'$ is a not-necessarily deterministic encryption function that takes as input a *public key* and a message and returns a ciphertext, and $\mathrm{Dec} : \mathcal{K}' \times \mathcal{X}' \rightarrow \mathcal{X}$ is a decryption function that takes as input a *private key* and a ciphertext and returns a message. A functional requirement is that if k_s is the private key corresponding to the public key k_p, $\mathrm{Dec}(k_s, \mathrm{Enc}(k_p, x)) = x$). The *IND-CPA game* for Enc is as follows:

1. A challenger picks a private key uniformly among all such keys for Enc, generates the corresponding public key and sends it to the adversary.
2. The adversary chooses two messages m_0, m_1 of equal length from the domain of Enc and sends them to the challenger.
3. The challenger picks $b \twoheadleftarrow \{0, 1\}$ and sends $\mathrm{Enc}(m_b)$ to the adversary.
4. The adversary returns $\hat{b} \in \{0, 1\}$ and wins iff. $\hat{b} = b$.

Let p be the winning probability of the adversary in the IND-CPA game (computed over all the samplings in the game, plus the ones possibly made by Enc and the adversary itself); the *IND-CPA advantage* of the adversary is defined to be $|2p - 1|$. Finally, the IND-CPA advantage $\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{Enc}}(t)$ of Enc is the maximum IND-CPA advantage of any adversary attacking Enc that runs in time t.*

Exercise 1: A game of martens and squirrels

The cute and clever pine marten (*martes martes*) is out in the woods hunting for the dim and pouchy grey squirrel (*sciurus carolinensis*). From a previous scouting mission, the marten knows that the squirrel sleeps every night in a different place, and it has carefully mapped all N such places. It also knows that the squirrel never sleeps twice in the same place until it has visited all the other $N - 1$ ones. The squirrel, aware of the presence of the marten, tries to always change the order in which it visits its sleeping grounds from one cycle to the other.

Q.1: The marten can visit T places per night to try catching the squirrel. Specify a simple hunting strategy for the marten that guarantees that it will catch the squirrel in at most $N - T$ nights.

Q.2: The marten being quite slender, it cannot reasonably expect to survive for $N - T$ nights (for typical values of N and T) without catching a squirrel. It is however highly skilled in scouting and astronomy, and so is always able to determine where the squirrel slept the *one* previous night and what is the day position in the current cycle (from 1 to N). Additionally, the squirrel —being quite silly— only uses a very primitive way of determining its sleeping schedule: at the start of every cycle, it picks $k \twoheadleftarrow [\![0, N - 1]\!]$ uniformly at random, and then decides that it will spend the i^{th} night of the cycle at place $i + k \mod N$ (where $a \mod b$ denotes here the unique non-negative remainder $\in [\![0, b - 1]\!]$ of the division of a by b), for some fixed numbering of the sleeping places (*i.e.* one that does not change from one cycle to another).

1. Assuming that the marten already knows the numbering of sleeping places used by the squirrel, give a strategy that guarantees that it will catch the squirrel within at most two nights.
2. Show that the former assumption is not necessary if the marten can spend one full cycle observing the squirrel (for instance because it is catching other squirrels in the meantime).

*Note that in a public-key setting, an adversary may (thanks to the knowledge of the public key) always itself encrypt messages of its choice without making any query to the challenger.

Q.3:

1. Show that the squirrel's strategy may be understood to implicitly use a block cipher, and give a general formulation thereof using an abstract "format-preserving" block cipher $E : \mathcal{K} \times [\![0, N-1]\!] \to [\![0, N-1]\!]$.

2. Show informally (*e.g.* using a reduction argument) that for this strategy, the marten's advantage in catching the squirrel is (among other things) a function of the PRP security of the squirrel's chosen block cipher.

Figure 1: Credit: wikimedia commons.

Exercise 2: Concatenation combiner

We define the hash function *concatenation combiner* $\mathrm{CAT}(F, G)$[†] as the map $x \mapsto F(x) || G(x)$. In other words, given two hash functions F and G, $\mathrm{CAT}(F, G)$ is the function that on input x gives as output the concatenation of $F(x)$ and $G(x)$.

Q.1: Let $H_1 : \{0,1\}^\star \to \{0,1\}^n$, $H_2 : \{0,1\}^\star \to \{0,1\}^n$ be two independent ideal hash functions, in that $\forall\, x \in \{0,1\}^\star, H_1(x) \twoheadleftarrow \{0,1\}^n, H_2(x) \twoheadleftarrow \{0,1\}^n$ (with all the drawings being independent). Show that $\mathrm{CAT}(H_1, H_2)$ is ideal (in the same sense).

Q.2: Let now H_1 be a "narrow-pipe" *Merkle-Damgård* hash function instantiated with an ideal compression function.

1. Recall the average complexity of computing a collision for H_2.

2. Recall an upper-bound on the average complexity of computing an N-multicollision for H_1.[‡]

3. Give a collision attack for $\mathrm{CAT}(H_1, H_2)$ with time cost $\Theta(n2^{n/2})$, and give its memory cost. (Hint: a step of the attack consists in computing an N-multicollision for H_1 for a well-chosen N).

4. Is $\mathrm{CAT}(H_1, H_2)$ still an ideal hash function?

[†] The horrendous signature of CAT is $(\{0,1\}^\star \to \{0,1\}^n) \times (\{0,1\}^\star \to \{0,1\}^{n'}) \to (\{0,1\}^\star \to \{0,1\}^{n+n'})$.

[‡] Recall that an N-multicollision for a hash function H is an N-uple $m_1, \ldots, m_N$ s.t. $H(m_1) = \cdots = H(m_N)$.

Q.3: Despite the previous attack, what do you think could be the interest (from a practical/engineering point of view) of using $\mathrm{CAT}(H_1, H_2)$ where H_1 and H_2 are narrow-pipe Merkle-Damgård.

Exercise 3: MACs and tags

In all of the following (independent) questions, we consider a deterministic MAC $M : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^n$, that is assumed to be "good".

Q.1: Suppose that $n = 128$, and let $t = t_L||t_R$ denote the output of M, where t_L (resp. t_R) are the 64 highest (resp. lowest) bits of t. We then consider the map $\mathrm{EXT} : \{0,1\}^{128} \to \{0,1\}^{192}$, $t_L||t_R \mapsto ((t_L \ll 1) \oplus t_R)||((t_L \gg 63) \oplus t_R)||((t_L \ggg 3) \oplus (t_R \lll 7))$, where $\ll$, $\gg$, $\lll$, $\ggg$, $\oplus$ respectively denote bitshift to the left, bitshift to the right, circular bitshift (or rotation) to the left, circular bit shift to the right and bitwise XOR.

1. Show that EXT is invertible.
2. What is the size of the proper image of EXT (that is the size of the set $\{x \in \{0,1\}^{192} \mid \exists\, y \in \{0,1\}^{128}, \mathrm{EXT}(y) = x\}$)?
3. Is there a cryptographic interest in defining M' from M as $M'(x) = \mathrm{EXT}(M(x))$?
4. Same question, if EXT were made "suitably non-invertible"?

Q.2: A certain network protocol authenticates every packet of 384 bits using M with $n = 96$. For every *session* of the protocol (what is a session is not important here, but in a typical day one expects 2^{40} sessions to be created worldwide), an identifier that is expected to uniquely identify the session among all possible sessions (past and future) is taken to be the 96-bit tag of a designated packet that is part of the session.

1. Explain why this overall process is badly-designed.
2. Propose a way to improve it.

Q.3: Suppose that M is a "good MAC" and is used in a challenge-response protocol (*e.g.* to grant access to a certain place or to authorise the usage of an object). For each of the following key/tag sizes (*i.e.* values for κ and n), describe one scenario where it would be an appropriate choice, or explain why there is none in your opinion.

1. $\kappa = 64$, $n = 64$.
2. $\kappa = 128$, $n = 64$.
3. $\kappa = 64$, $n = 128$.
4. $\kappa = 256$, $n = 256$.

Exercise 4: Textbook PKC

In the following questions, $\mathbb{G}$ is a finite commutative cyclic group of prime order p (meaning that it contains p elements), and g is a publicly-known generator of $\mathbb{G}$. You may assume that basic arithmetic in $\mathbb{G}$ (computing a product, an exponentiation and an inversion) is "efficient".

You must carefully justify all your answers in this exercise, however *no formal proofs are expected.*

Q.1: 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 $a, b \twoheadleftarrow [\![0, p-1]\!]$ or $(g^a,\ g^b,\ g^c)$, with $a, b, c \twoheadleftarrow [\![0, p-1]\!]$, each with probability 0.5. The adversary wins if it correctly guesses which triple it was given.

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

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

Q.2:

1. Explain how the receiver can decrypt a message sent by the sender.
2. Show informally 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"?

Q.3: Assuming that one wishes for the above scheme to reach the best possible IND-CPA security, explain why it is important that the sender picks b uniformly (and independently for every message).

Q.4: As an attacker, you are recording all the messages that are being sent (by possibly several senders) to a given receiver.

1. Suppose that the same value for b is used for two different messages. Explain what information you could deduce as a result.
2. Explain why textbook Elgamal cannot informally be said to be "beyond-birthday secure".
3. Given what you know about the hardness of computing discrete logarithms in a finite commutative cyclic group, is the previous attack a limiting factor for the security of textbook Elgamal?

Q.5: Suppose you wish to secretly communicate with someone over the internet, and want to rely on textbook Elgamal (with well-chosen parameters). Would this encryption scheme alone provide adequate security, or would some additional cryptographic primitives or constructions be needed?