Cryptography and Security — Final Exam

Serge Vaudenay

19.1.2024

- duration: 3h
- no documents allowed, except one 2-sided sheet of handwritten notes
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **not** answer any technical question during the exam
- readability and style of writing will be part of the grade
- answers should not be written with a pencil

1 The Security of Nonce-Based Symmetric Encryption

We recall the CPCA decryption game against non-based symmetric encryption on message space $\mathcal{D}$, nonce space $\mathcal{N}$, and key space $\{0,1\}^k$.

Game
1: $K \xleftarrow{\$} \{0,1\}^k$
2: $X_0 \xleftarrow{\$} \mathcal{D}$, $N_0 \xleftarrow{\$} \mathcal{N}$
3: $\mathsf{Used} \leftarrow \{N_0\}$
4: $Y_0 \leftarrow \mathsf{Enc}(K, N_0, X_0)$
5: $\mathcal{A}^{\mathsf{OEnc},\mathsf{ODec}}(N_0, Y_0) \to X$
6: **return** $1_{X=X_0}$

Oracle $\mathsf{OEnc}(N, X)$:
7: **if** $N \in \mathsf{Used}$ **then return** $\perp$
8: $\mathsf{Used} \leftarrow \mathsf{Used} \cup \{N\}$
9: **return** $\mathsf{Enc}(K, N, X)$

Oracle $\mathsf{ODec}(N, Y)$:
10: **if** $(N, Y) = (N_0, Y_0)$ **then return** $\perp$
11: **return** $\mathsf{Dec}(K, N, Y)$

Q.1 List the differences with the CPA decryption game. Do the same for the CPCA key recovery game and the CPA key recovery game.

Q.2 Recall how to adapt the Vernam cipher to make a nonce-based symmetric encryption scheme with a PRNG, and show that it is insecure against CPCA decryption attacks.

Q.3 We change the CPA decryption attack game by allowing nonce reuse in the encryption oracle. Show that a nonce-based symmetric encryption scheme based on the Vernam cipher is insecure in this game.

Q.4 We use a block cipher C_K and assume that the block space is given a finite field structure $\mathbf{F} = \mathcal{D}$. We define $\mathsf{Enc}(K, N, \mathsf{pt}) = \mathsf{pt} \times K + C_K(N)$, for $K \in \mathbf{F}^*$ and $\mathsf{pt} \in \mathbf{F}$. Show that this nonce-based symmetric encryption system is insecure against CPCA key recovery attacks.

Q.5 For a stateless implementation of the message sender, what would you recommend for a nonce length in practice and why? (Here, stateless means that the sender does not keep more than the secret key in memory in between two messages to send.)

2 Privacy Pass

We consider a group G of prime order q. We use additive notations for this group. Let H be a random function from the set of bistrings to $G - \{0\}$. We assume that H is publicly known. We consider a function family $(f_K)_{K \in \mathbf{Z}_q^*}$ from the set of bitstrings to G defined by $f_K(x) = K \cdot H(x)$. We call K a secret key. We design a token-issuance protocol between a client and a server as follows:

- The client has a bitstring x as input. The server has a key K as input.
- The client picks $\lambda \in \mathbf{Z}_q^*$, sets $X = \lambda \cdot H(x)$, and sends X to the server.
- The server computes $Y = K \cdot X$ and returns it to the client.
- The client computes $Z = \frac{1}{\lambda} \cdot Y$ and takes it as an output.

A token is a (x, Z) pair. After the token is issued by the above protocol, the client can redeem the token to the server. The server accepts the token as valid if it satisfies $Z = f_K(x)$. We consider the following security game:

Game Γ_b
1: pick $H : \{0,1\}^* \to G - \{0\}$
2: $K \xleftarrow{\$} \mathbf{Z}_q^*$
3: pick $F : \{0,1\}^* \to G - \{0\}$
4: $\mathcal{A}^{\mathsf{OH},\mathsf{OF}} \to z$
5: **return** z

Oracle $\mathsf{OH}(x)$:
6: **return** $H(x)$

Oracle $\mathsf{OF}(x)$:
7: **if** $b = 0$ **then return** $F(x)$
8: **return** $f_K(x)$

The advantage of adversary $\mathcal{A}$ is $\mathsf{Adv} = \Pr[\Gamma_1 \to 1] - \Pr[\Gamma_0 \to 1]$. We assume that this construction is secure in the sense that for any adversary $\mathcal{A}$ limited to a *feasible* complexity and number of oracle queries, the advantage is *negligible*.

Q.1 If we remove the first line of the game and the oracle OH, what is this security notion? Why did we add the oracle OH?

Q.2 Let x be a random bitstring following an arbitrary distribution, H be an arbitrary function (fixed), K be the random key, and λ be the random mask. In the issuance protocol, prove that X is independent from x.

Q.3 The token-issuance protocol is sometimes called an *oblivious* evaluation protocol for $f_K(x)$. Explain (guess) why?

Q.4 Assume that a malicious client can interract with a token-issuance-server oracle and with a token-redeem-server oracle. We want to formalize the security notion saying that if the adversary interacts n times with the issuance server oracle, then it cannot produce $n + 1$ pairwise different valid tokens. This is called one-more unforgeability (OMUF). Propose a game to define OMUF security.

3 Discrete Log in a Small Set

We consider a group G of large prime order q with a generator g. Given $y \in G$, the discrete logarithm problem is the problem of finding $x \in \mathbf{Z}_q$ such that $y = x \cdot g$. One generic algorithm to solve that is the Baby-Step Giant-Step algorithm which is recalled below.

Input: g and y in a group G, B an upper bound for $\#G$
Precomputation (y not provided)
1: set $\ell = \lceil \sqrt{B} \rceil$
2: **for** $i = 0, \ldots, \ell - 1$ **do**
3: define $T\{(i\ell) \cdot g\} = i$
4: **end for**
Algorithm (y provided)
5: **for** $j = 0, \ldots, \ell - 1$ **do**
6: compute $z = y - j$
7: **if** $T\{z\}$ is defined **then**
8: yield $x = \ell T\{z\} + j$ and stop
9: **end if**
10: **end for**

We let S be a small subset of $\mathbf{Z}_q$ consisting of all integers between a and b, with $b - a$ small. That is, $S = \{a, a + 1, \ldots, b\}$. We call S a small interval of $\mathbf{Z}_q$. Given $y \in G$, we consider the problem of finding $x \in \mathbf{Z}_q$ such that $y = x \cdot g$, when we know that there exists one such x in S.

Q.1 Optimize the implementation of the Baby-Step Giant-Step in order to minimize the total number of group additions. Analyze the precomputation complexity (total number of group additions in precomputation), time complexity (total number of group additions in the algorithm using y), and memory complexity (total number of entries defined in T). (Give the worst case and the average case.)

Q.2 Propose a variant of this algorithm for the problem in this exercise (with solutions in S) and analyze its precomputation complexity, time complexity, and memory complexity.

Q.3 What time-memory tradeoffs are possible in the algorithm?

Q.4 Propose a way to adapt the ElGamal encryption to encrypt a *small* bitstring instead of a group element. Specify the size of this plaintext.