Cryptography and Security — Final Exam

Serge Vaudenay

28.1.2021

- 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

1 Merkle's Puzzles

We define a primitive which we call *puzzle* with the following algorithms:

- $\mathsf{Gen}(s, \lambda, i, k; r) \to p$ is probabilistic and generates a "puzzle" p from some input index i, a key k, a difficulty s, a parameter λ, and random coins r. We call a unit of time its average time complexity.
- $\mathsf{Solve}(s, \lambda, p) \to (i, k)$ is deterministic and recovers i and k from s, λ, and p. Its time complexity is upper bounded by 2^s units of time.

The correctness notion says

$$\forall s, \lambda, i, k \qquad \Pr[\mathsf{Solve}(s, \lambda, \mathsf{Gen}(s, \lambda, i, k)) \to (i, k)] \geq 1 - 2^{-\lambda}$$

The security notion intuitively says that any solving algorithm requires at least an average complexity of $\frac{1}{2}2^s$ units of time.

Let Enc be a symmetric encryption scheme over the space of (i, k, z) tuples (where z is a bitstring of length $s + \lambda$) with a key of length at least s. We assume that encryption and decryption take approximately the same amount of time. Given s random coins r, we define $\mathsf{Gen}(s, \lambda, i, k; r) = \mathsf{Enc}_{r\|0^*}(i, k, 0^{s+\lambda})$, where $r\|0^*$ is r padded with enough zero bits to be a key of correct length and $0^{s+\lambda}$ is a string of $s + \lambda$ zero bits.

Q.1 Prove that we can define a puzzle based on Gen (i.e. propose a Solve and verify all conditions of a puzzle).

Q.2 Let s, λ, and N be three public parameters. Alice generates $\mathsf{Gen}(s, \lambda, i, k_i; r_i) \to p_i$ for $i = 1, \ldots, N$ and randomly selected k_i and r_i. Alice picks a random permutation σ of $\{1, \ldots, N\}$, defines $p'_j = p_{\sigma(j)}$, $i = 1, \ldots, N$, and sends $p'_1, \ldots, p'_N$ to Bob. Bob picks a random j, runs $\mathsf{Solve}(s, \lambda, p'_j) \to (i, k_B)$ and sends i back to Alice. Alice sets $k_A = k_i$. The private outputs of Alice and Bob are k_A and k_B respectively.
Prove that we have a key agreement protocol (i.e. prove correctness). What is the complexity for Alice and Bob?

Q.3 In Q.2, what is the best passive attack Eve could do? (Study its complexity.) As s and N are based on the security parameter, propose a way to select them coherently.

2 Crypto Choices

We want to design a secure communication system in which breaking it would be equivalent, in terms of complexity, to do an exhaustive search on a 128-bit secret key. In this exercise, we have to *propose* concrete cryptographic algorithms with some concrete parameters (such as the length of various objects). (*Proposing* can mean to name an existing standard algorithm instance which could be used with brief justifications.) To get the full grade, we should propose parameters which are neither too small not too large.

Q.1 Propose a symmetric encryption scheme for data of variable length. Specify the key length. Which security property is this algorithm supposed to guarantee?

Q.2 For a message authentication code, what would be the appropriate key length and tag length? Propose an algorithm with those parameters. Which security property is this algorithm supposed to guarantee?

Q.3 We want to hash a message to sign its digest. Propose an appropriate hash function. Which security property is important when used in hash-and-sign?

Q.4 Propose a signature scheme which is appropriate with the proposed hash function.

Q.5 Revisit all questions to offer the same security in a post-quantum era.

3 ECDSA with Bad Randomness

We recall the ECDSA scheme. A point G in an elliptic curves generates a group of order a public prime n. The secret key is a residue d modulo n. The public key is a point $Q = dG$. To sign a message M, the signer picks a random $k \in \mathbf{Z}_n^*$ and computes $r = \mathsf{xcoord}(kG) \bmod n$ and $s = \frac{H(M)+dr}{k} \bmod n$, where H is a hash function. The signature is the pair (r, s). The signature is such that

$$r = \mathsf{xcoord}\left(\frac{H(M)}{s}G + \frac{r}{s}Q\right)$$

We consider a network of signers where the signer of index i has a key pair (d_i, Q_i). Signers use devices to sign but those devices have a poor random generator. We assume that there exists a subset of $\mathbf{Z}_n^*$ of cardinality v such that the random generator outputs k uniformly in this subset. We assume that n has 256 bits and that v has 100 bits.

Q.1 Describe a key recovery attack targetting one user. Analyze how many signatures are needed, the complexity, and the probability of success. Is this attack realistic?

Q.2 In a network of N signers, we assume that each signer signs w documents at random. We assume $w \ll \sqrt{v}$. Given two different signers, what is (approximately) the probability p that they once used the same k value?

Q.3 If each signer signs w documents, design an attack which recovers the key of some signers with good probability when $N \sim vw^{-2}$. Apply this to $w \sim 2^{35}$.
HINT: In a random graph of N vertices in which each possible edge is present with probability $\frac{1}{N}$, there exists a cycle with good probability.

4 Post-Quantum Bitcoins Wallet

In the bitcoin architecture, a transaction is an ECDSA signature by a holder of a key pair $(\mathsf{pk}, \mathsf{sk})$ on a document which contains

– pk;
– a list $p_1, \ldots, p_m$ of pointers to some transactions in which p_i paid x_i bitcoins to pk which have not been spent so far;
– a list $(y_1, \mathsf{pk}_1), \ldots, (y_n, \mathsf{pk}_n)$ of public keys to whom to pay the collected bitcoins and of positive numbers y_j such that $x_1 + \cdots + x_m = y_1 + \cdots + y_n$.

When a user wants to pay some other users for a total of $y_1 + \cdots + y_{n-1}$, he/she collects some of his bitcoins $x_1 + \cdots + x_m$ so that $x_1 + \cdots + x_m \geq y_1 + \cdots + y_{n-1}$, he/she computes y_n such that $x_1 + \cdots + x_m = y_1 + \cdots + y_n$ and if $y_n > 0$, he/she adds (y_n, pk) to pay the leftover to himself.

Q.1 In a quantum era, show that as soon as the transaction is made public, a powerful adversary can steal the $y_1 + \cdots + y_n$ bitcoins.

Q.2 We change the structure of the transaction so that the (y_j, pk_j) pairs in the list of the recipients of the transaction are replaced by some $(y_j, H(\mathsf{pk}_j))$ pairs, by using a one-way hash function H. What users make public is now $H(\mathsf{pk})$, but users reveal their pk in the transaction. Using a notion of "one-time secret key" sk, show that we can mitigate the previous attack.

Q.3 Describe a structure of a wallet in which each user would have a unique secret K.