Advanced Cryptography — Final Exam

Serge Vaudenay

27.6.2013

- duration: 3h00
- any document is allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will not answer any technical question during the exam
- the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

1 ElGamal using a Strong Prime

Let p be a large strong prime. I.e., p is a prime number and $q = \frac{p-1}{2}$ is prime as well.

Q.1 Show that QR_p is a cyclic group.

Q.2 Show that -1 is not a quadratic residue modulo p.

Q.3 Show that there exists a bijection σ from $\{1,\ldots,q\}$ to QR_p, the group of quadratic residues in Z_p^*, such that for all x, $\sigma(x) = x$ or $\sigma(x) = -x$.

Q.4 For $m \in \{1,\ldots,q\}$ and $x \in \mathsf{QR}_p$, give algorithms to compute $\sigma(m)$ and $\sigma^{-1}(x)$.

Q.5 We consider the following variant of the ElGamal cryptosystem over the message space $\{1,\ldots,q\}$. Let g be a generator of QR_p. The secret key is $x \in \mathbf{Z}_{p-1}$. The public key is $y = g^x \bmod p$. To encrypt a message m, we pick $r \in \mathbf{Z}_{p-1}$, compute $u = g^r \bmod p$, and $v = \sigma(m)y^r \bmod p$. The ciphertext is the pair (u, v).
Describe the decryption algorithm.

Q.6 Show that this variant is IND-CPA secure when the DDH problem is hard in QR_p.

2 BLS Signature

Let p be a prime number, G and G_T be two groups (with multiplicative notations) of order p, g be a generator of G, and e be a function from $G \times G$ to G_T such that

- (non-degenerate) there exists $a, b \in G$ such that $e(a, b) \neq 1$;
- (efficiently computable) e can be evaluated efficiently;
- (bilinear) $e(ab, c) = e(a, c)e(b, c)$ and $e(a, bc) = e(a, b)e(a, c)$ for all $a, b, c \in G$.

We assume that the size of p is polynomially bounded. We assume that we have efficient algorithms for group multiplication (in both groups), as well as for comparing group elements. We assume that a random oracle H maps any bitstring to a group element in G. We define a signature scheme as follows:

key generation: we pick the secret key $x \in \mathbf{Z}_p$ and the public key is $v = g^x$;
signature algorithm: to sign a message m, we produce $\sigma = H(m)^x$;
verification algorithm: to verify (v, m, σ), we check that $e(g, \sigma) = e(v, H(m))$.

Q.1 Show that $e(g^x, g^y) = e(g, g)^{xy}$ for all $x, y \in \mathbf{Z}_p$.

Q.2 Show that the algorithms in the signature scheme are efficient and that produced signatures are always correct.

Q.3 Show that the Decisional Diffie-Hellman (DDH) problem is easy to solve in G.

Q.4 For an attack using no chosen message, show that making an existential forgery implies solving the Computational Diffie-Hellman (CDH) problem. More precisely, given an algorithm $\mathcal{A}^H(g, v) = (m, \sigma)$ forging a valid signature σ for m under public key v with oracle access to H, we can construct an algorithm $\mathcal{B}(g, g^x, Y)$ to compute Y^x, with complexity comparable to the one of $\mathcal{A}$ and a polynomially bounded overhead. (Assume $\mathcal{A}$ works with probability 1.)
Hint: simulate $H(m')$ by $g^{r(m')}Y$ where r is a random function from $\{0, 1\}^*$ to $\mathbf{Z}_p$.

Q.5 If now $\mathcal{A}$ works with probability ρ over the uniform distribution of X and H in G, show that we can construct some $\mathcal{B}'$ working with probability ρ as well, for any x and y.

Q.6 Show that by selecting a biased function s from $\{0, 1\}^*$ to $\{0, 1\}$ and by now simulating H by $H(m') = g^{r(m')}Y^{s(m')}$, we can introduce chosen message attacks in the previous result: making existential forgeries under chosen message attacks implies solving the CDH problem. (The probability of the solving algorithm may be different though.)

3 PRF Programming

A function $\delta(s)$ is called negligible and we write $\delta(s) = \mathsf{negl}(s)$ if for any $c > 0$, we have $|\delta(s)| = o(s^{-c})$ as s goes to $+\infty$.

Let s be a security parameter. For simplicity of notations, we do not write s as an input of games and algorithms but it *is* a systematic input.

A family $(f_k)_{k\in\{0,1\}^s}$ of functions f_k from $\{0,1\}^s$ to $\{0,1\}^s$ is called a PRF (Pseudo Random Function) if for any probabilistic polynomial-time oracle algorithm $\mathcal{A}$, we have that

$$|\Pr[\mathcal{A}^{f_K(\cdot)} = 1] - \Pr[\mathcal{A}^{f^*(\cdot)} = 1]| = \mathsf{negl}(s)$$

where $K \in \{0,1\}^s$ is uniformly distributed, f^* is a uniformly distributed function from $\{0,1\}^s$ to $\{0,1\}^s$, $f_K(\cdot)$ denotes the oracle returning $f_K(x)$ upon query x, and $f^*(\cdot)$ denotes the oracle returning $f^*(x)$ upon query x.

Given a PRF $(f_k)_{k\in\{0,1\}^s}$, we construct a family $(g_k)_{k\in\{0,1\}^s}$ by $g_k(x) = f_k(x)$ if $x \neq k$ and $g_k(k) = k$. The goal of the exercise is to prove that $(g_k)_{k\in\{0,1\}^s}$ is a PRF.

We define the PRF game played by $\mathcal{A}$ for g, f, and f^* by

Game Γ^g
pick $K \in \{0,1\}^s$
run $b = \mathcal{A}^{g_K(\cdot)}$
give b as output

Game Γ^f
pick $K \in \{0,1\}^s$
run $b = \mathcal{A}^{f_K(\cdot)}$
give b as output

Game Γ^*
pick $f^* : \{0,1\}^s \to \{0,1\}^s$
run $b = \mathcal{A}^{f^*(\cdot)}$
give b as output

For each integer i, we define an algorithm $\mathcal{A}_i$ (called a *hybrid*) which mostly simulates $\mathcal{A}$ until it makes the ith query. More concretely, $\mathcal{A}_i$ simulates every step and queries of $\mathcal{A}$ while counting the number of queries. When the counter reaches the value i, $\mathcal{A}_i$ does not make this query k but it stops and the queried value k is returned as the output of $\mathcal{A}_i$. If $\mathcal{A}$ stops before making i queries, $\mathcal{A}_i$ stops as well, with a special output $\perp$. We define the following games:

Game Γ_i^f
pick $K \in \{0,1\}^s$
run $k = \mathcal{A}_i^{f_K(\cdot)}$
if $k = \perp$, stop and output 0
pick $x \in \{0,1\}^s$
if $f_k(x) = f_K(x)$, stop and output 1
output 0

Game Γ_i^*
pick $f^* : \{0,1\}^s \to \{0,1\}^s$
run $k = \mathcal{A}_i^{f^*(\cdot)}$
if $k = \perp$, stop and output 0
pick $x \in \{0,1\}^s$
if $f_k(x) = f^*(x)$, stop and output 1
output 0

Let $F(\Gamma)$ be the event that any of the queries by $\mathcal{A}$ in game Γ equals K. We assume that the number of queries by $\mathcal{A}$ is bounded by some polynomial $P(s)$.

Q.1 Show that $|\Pr[\Gamma^f \to 1] - \Pr[\Gamma^* \to 1]| = \mathsf{negl}(s)$.
Q.2 Show that $\Pr[\Gamma^g \to 1|\neg F(\Gamma^g)] = \Pr[\Gamma^f \to 1|\neg F(\Gamma^f)]$ and $\Pr[\neg F(\Gamma^g)] = \Pr[\neg F(\Gamma^f)]$.
Q.3 Deduce $|\Pr[\Gamma^g \to 1] - \Pr[\Gamma^f \to 1]| \le \Pr[F(\Gamma^f)]$.
Q.4 Show that $\Pr[F(\Gamma^f)] \le \sum_{i=1}^{P(s)} \Pr[\Gamma_i^f \to 1]$.
Q.5 Show that $|\Pr[\Gamma_i^f \to 1] - \Pr[\Gamma_i^* \to 1]| = \mathsf{negl}(s)$ for all $i \le P(s)$.
Q.6 Show that $\Pr[\Gamma_i^* \to 1] = \mathsf{negl}(s)$ for all $i \le P(s)$.
Q.7 Deduce $|\Pr[\Gamma^g \to 1] - \Pr[\Gamma^* \to 1]| = \mathsf{negl}(s)$.