National Technical University of Athens
School of Electrical and Computer Engineering

Computational Cryptography

(SIMMY, SEMFE, ALMA, EME, MP)

3rd Series of Exercises

Deadline for submission: Before the course exam

Exercise 1. A company manager needs to frequently receive encrypted messages from his employees. For this purpose, he uses RSA, that is, he gives everyone his public key $\langle n, e\rangle$ where $n = pq$ with p, q primes. Of course it keeps the primes p, q hidden.
For convenience, he also gives his secretary a device with which employees who do not have the encryption program can encrypt their messages.
The device is supposed to work as follows for input m:

- Calculates $c_p = m^e \bmod p$,
- Computes $c_q = m^e \bmod q$, and
- Combines the solutions with CRT to output the unique value $c \in Zed_n$ t.o. $c \equiv m^e \pmod n$.

However, due to a manufacturing error, in the second step the device calculates $c'_q = m^e + 1 \bmod q$ and gives the output $c' \in \mathbb{Z}_n$, t.o. $c' \equiv c_p \pmod p$ and $c' \equiv c'_q \pmod q$.
Naturally, the manager soon realizes (how?) that something is wrong, and asks by his secretary to send the device for repair. However, the secretary, who has taken cryptography courses, manages to find the director's private key before sending the device to the service. How did she manage this?

Exercise 2. [KL2, Exercise 8.20] Let KGen the RSA key generator (as in the relevant slides, p. 10). We construct a hash function as below.

- $(N, e, d) \leftarrow KGen(1^n), \quad y \in_R \mathbb{Z}_N^*, \quad s = (N, e, y)$.
- With $f_b^s(x) = y^b x^e \mod N$ for $b \in \{0, 1\}$, we define $H^s(x) : \{0, 1\}^{3n} \to \mathbb{Z}_N^*$ as follows:

$$H^s(x) = f^s_{x_1}(f^s_{x_2}(\cdots(1)\cdots)) \qquad (3n \text{ functions}).$$

Show that if the RSA problem is hard, then H is collision-resistant. Specifically, if with non-negligible probability one can find $u, v \in \{0,1\}^{3n}$ with $u \neq v$ and $H^s(u) = H^s(v)$, then the RSA problem is not difficult for KGen.

Exercise 3. Let $\mathbb{G}$ be a cyclic group with prime order q and generator g.

1. We define the Quadratic Diffie - Hellman (SDH) problem as follows:
   Given $g, g^x \in \mathbb{G}$. Compute g^{x^2}.
2. We define the Inverse Diffie - Hellman (IDH) problem as follows:
   Given $g, g^x \in \mathbb{G}$. Compute $g^{x^{-1}}$.

Prove that the above problems are equivalent to each other and to the computational Diffie - Hellman (CDH) problem.

Exercise 4. Implement in a programming language of your choice the RSA decryption attack on a ciphertext c that uses an oracle that can decide whether the message corresponding to the ciphertext is in the 'upper' or 'lower' half of $\mathbb{Z}_n$ (i.e. the loc function - see RSA slides).

Specifically, you must implement 2 pieces of code:
(1) The first will 'simulate' the oracle (normally decrypting with the private key) c and computing loc.
(2) The second will implement the attack by repeatedly asking the oracle appropriate questions about the loc.

For the communication of the programs you can use any form of interprocess communication (RPC) you know, or even simpler communication via a file or internally in the program with an appropriate function call. The generation of the keys and the initial encryption can be done by your own code or using a ready-made tool like Openssl.

Exercise 5. In the lecture on Discrete Logarithm (DL) cryptosystems we saw that leaking the DL parity via the Legendre symbol can allow the construction of an efficient distinguisher for Diffie-Hellman triples and consequently the solution of the DDH problem in $\mathbb{Z}_p^*$. Can this specific leak lead to the disclosure of the entire discrete logarithm in $\mathbb{Z}_p^*$? Justify your answer.

Exercise 6. Given the following variant of the ElGamal protocol which operates on a group $\mathbb{G}$ with order prime q and generators g, h:

- $\mathsf{KGen}(\lambda) \rightarrow (\mathsf{sk}, \mathsf{pk}) = (x, g^x)$ where $x \leftarrow\$ \mathbb{Z}_q^*$
- $\mathsf{Enc}(\mathsf{pk}, m) \rightarrow (\mathsf{pk}^r, g^r h^m)$ where $r \leftarrow\$ \mathbb{Z}_q^*$

1. Define the decryption function $\mathsf{Dec}(\mathsf{sk}, m)$ and prove its correctness.

2. Study the security of the variant with respect to the properties OW-CPA, IND-CPA, IND-CCA.

Assume that the parameter generation process has been done honestly (i.e., that g, h are uniformly chosen generators with unknown distinct logarithms).

Exercise 7. Given a group $\mathbb{G}$ of order q prime where the Diffie - Hellman decision problem is hard. Let g, h be generators of $\mathbb{G}$ such that the distinct logarithms $\log_g h$ and $\log_h g$ are unknown. On this group we define the Pedersen commitment scheme with commitment algorithm $\mathsf{Commit}(m, r) = c = g^m h^r$ with $m, r \in \mathbb{Z}_q^*$.
Consider the following protocol Π with public input $\langle \mathbb{G}, g, h, q, c \rangle$ that proves that the prover knows m, r such that $c = g^m h^r$:

- The prover uniformly chooses a $t_1, t_2 \in \mathbb{Z}_q^*$ and sends to the verifier to $t = g^{t_1} h^{t_2}$.
- The verifier uniformly chooses $e \in \mathbb{Z}_q^*$ and sends it to the prover.
- The prover computes $s_1 = t_1 + em \bmod q$, $s_2 = t_2 + er \bmod q$ and sends them to the verifier.
- The verifier accepts if and only if $g^{s_1} h^{s_2} = t c^e$.

1. Is Π a Σ-protocol, i.e. does it have completeness, special correctness, honest verifier zero-knowledge? Justify your answers.
2. Is Π witness indistinguishable? That is, given an honest prover and a common public input $\langle \mathbb{G}, g, h, q, c \rangle$ what conclusions can a malicious verifier draw from the conversations (t, e, s_1, s_2) for witness (m, r) and (t', e', s_1', s_2') for witness (m', r') with $m \neq m'$ and $r \neq r'$.
3. We change Π to Π', so that in the first step the prover calculates and sends instead of $t = g^{t_1} h^{t_2}$ the values $a = g^{t_1}$ and $b = h^{t_2}$. What is the relation that the verifier must check to be convinced that the prover knows m, r? Is Π' now a Σ-protocol?

Exercise 8. Given the following protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ which aims to prove knowledge of the message corresponding to a given RSA ciphertext with public key (e, n), namely $m \in \mathbb{Z}_n^*$ such that $y = m^e \bmod n$. Furthermore, assume that e is prime.

- $\mathcal{P}$ randomly chooses a $t \in \mathbb{Z}_n^*$ and sends $\mathcal{V}$ $h = t^e \bmod n$.
- $\mathcal{V}$ randomly chooses a c, $c \in \{0, \ldots, e-1\}$, and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $r = t m^c \bmod n$ and sends it to $\mathcal{V}$.
- $\mathcal{V}$ accepts if and only if $r^e \equiv h y^c \pmod{n}$.

Prove that the above is a Σ-protocol. For the HVZK property, the proof should be at the level of analysis followed in the slides, but the protocol transcripts and the probability of their occurrence should be shown in detail.

Exercise 9. In the non-interactive version of the Schnorr protocol (see ZK lecture) the challenge is computed as $c := \mathsf{H}(h||y)$, where y is the prover's first message and h is the prover's public key. Let be the variant with $c := \mathsf{H}(y)$, i.e. without including the prover's public key. Prove that this version does not have the correctness property.

Hint: Design an attack where a malicious prover can construct a valid proof of knowledge of the discrete logarithm *of some* element of the group $\mathbb{G}$ without knowing it.

Exercise 10. Implement the Schnorr signature scheme in a programming language of your choice. For Schnorr signatures, use a first-order subgroup q of $\mathbb{Z}_p^*$. The implementation should also verify the Schnorr signature on some data of your choice.

Exercise 11. Let be the following signature scheme where the parameters are the same as in the ElGamal signature scheme. Each user has a private key x and a public key $y = g^x \bmod p$. The signature works as follows:

i. The signer initially chooses $h \in \{0, \ldots, p{-}2\}$ such that: $\mathsf{H}(m) + x + h \equiv 0 \pmod{p-1}$, where H is a collision resistant hash function.
ii. The signature is the triple: $\mathsf{Sign}(x, m) = (m, (x + h) \bmod p - 1, g^h \bmod p)$.
iii. To verify that a triple (m, a, b) is a valid signature, it is checked if:

- $yb \equiv g^a \pmod{p}$ and
- $g^{\mathsf{H}(m)} yb \equiv 1 \pmod{p}$.

Show that this scheme does not protect against a universal forgery attack.

In all exercises, we denote XOR by "$\oplus$" and the juxtaposition by "$||$".

Short instructions: (a) try it yourself, (b) discuss with your classmates, (c) search for ideas on the Internet – in that order and after dedicating enough time to each stage! In any case, the answers should be *strictly individual*. You may be asked to briefly present some of your solutions.

Good luck!