# Introduction to Cryptography – LMAT2450
# Final Examination

January 5, 2015

### Instructions

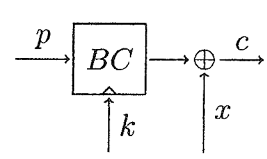
1. You can use the slides presented during the class, and all your personal notes. No book or other printed/photocopied material is allowed.

2. The duration of the exam is 3 hours. Answer the questions on ***separate*** sheets of paper.

3. You have the possibility to present your answers to the examiners.

**Question 1** We are interested in the construction of hash functions $\langle \mathsf{Gen}, H \rangle$ from a block cipher. Our constructions make use of a simplified version of the Merkle-Damgård transform, that is, we hash a message $m = m_1, \ldots, m_l$ (where each $m_i$ is $n$ bits long) by computing $h_i = F(m_i, h_{i-1})$, using a public constant (say, 0) as $h_0$ and $h_l$ as output of the hash function (note that we do not add an extra block containing the message length.)

The $F$ function is built using a block cipher $BC$ (with $n$ bits block size and key size,) as depicted in the figure below, where $p, k$ and $x$ can take any of the following four values: $m_i$, $h_{i-1}$, $m_i \oplus h_{i-1}$ and $s$ where $s$ is a public random bit-string produced by $\mathsf{Gen}$. The output of $F$ is always taken as $c$.




In total, we have 64 possible ways of building $F$: 3 variables can take 4 possible values. Those 64 hash function candidates have been studied in detail in the literature, and only 12 among them provide a secure hash function. We ask you to show how each of the 3 candidates below fail to provide one of the 3 standard security properties of a hash function (try to break the weakest possible property.)

1. $p = h_{i-1}$, $k = s$, $x = m_i$ (that is, $F(m_i, h_{i-1}) = BC_s(h_{i-1}) \oplus m_i$).
2. $p = s$, $k = m_i$, $x = h_{i-1}$.
3. $p = s$, $k = m_i \oplus h_{i-1}$, $x = s$.

**Question 2** Blum and Goldwasser proposed a public key encryption scheme defined as follows.

- $\mathsf{Gen}(1^n)$ picks two random primes $p$ and $q$ of length $n$ and equal to 3 mod 4. It outputs the public key $N = pq$ and the private key $(p, q)$. For such choices of $p$ and $q$, there is an efficient function *sqrt* that, given $p$ and $q$ and an element of $QR_N$ (i.e., an element of $\mathbb{Z}_N^*$ that is the square of another element of $\mathbb{Z}_N^*$), computes the single square root of this element that is itself in $QR_N$.

- $\mathsf{Enc}_N(m)$ parses $m$ as $l$ bits $m_1, \ldots, m_l$. Then it picks a random $a_0 \to \mathbb{Z}_N^*$ and computes the sequence $a_1, \ldots, a_{l+1}$ by successive squaring, that is, $a_i = a_{i-1}^2 \bmod N$. The ciphertext $c$ is computed as $(a_{l+1}, m_1 \oplus LSB(a_1), \ldots, m_l \oplus LSB(a_l))$, where $LSB$ is the function that extracts the least significant bit of its input.

1. Explain how the $\mathsf{Dec}$ algorithm works for this scheme. (You can use the *sqrt* function as a black box.)

2. Prove that the Blum-Goldwasser encryption scheme is not IND-CCA secure.

The SRLSB game, defined as follows, has been proven as difficult to win as factoring $N$:

$\mathcal{A}$ wins the SRLSB game if it guesses $LSB(a)$ from the outputs of the following experiment:

a) Run $\mathsf{Gen}$ as defined above and output the modulus $N$.
b) Pick a random element $a \in QR_N$ (by picking a random element in $\mathbb{Z}_N^*$ and squaring it) and output $a^2 \bmod N$.

3. Consider a restricted version of the Blum-Goldwasser encryption scheme where the message space is $\{0, 1\}$ (that is, $l = 1$). Prove that this scheme is IND-CPA secure under the SRLSB assumption.

4. Extend your previous result to the case of messages of arbitrary length that differ by their first bit only.

**Question 3** Your answer to each of the following questions should take less than 5 lines.

1. Let $\Pi = \langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be an IND-CPA secure encryption scheme with plaintext space $\mathcal{M}$ and ciphertext space $\mathcal{C}$. Suppose that the plaintext space can be partitioned into $\mathcal{P}_1$ and $\mathcal{P}_2$ of equal size and that there exists a function $f : \mathcal{C} \to \{0, 1\}$ so that $f(c) = 1$ iff $c$ is the encryption of a message in $\mathcal{P}_1$. Prove that, if $\Pi$ is IND-CPA, then no adversary can efficiently compute $f$.

2. Let $G : \{0,1\}^* \to \{0,1\}^*$ be a pseudorandom generator with 1 bit expansion (that is, $\forall s \in \{0,1\}^*$, it holds that $|G(s)| = |s| + 1$). Show that this pseudorandom generator is insecure in front of a computationally unbounded adversary.

3. Alice computes Pedersen commitments on (secret) messages $m_1$ and $m_2$ as $c_1 = g^{m_1} h^{r_1}$, $c_2 = g^{m_2} h^{r_2}$ (computing in a cyclic group of prime order $q$, and with $r_1$ and $r_2$ being random values in $\mathbb{Z}_q$, as usual) and wants to prove that she knows an opening on the commitment $c_3 = c_1^{m_2} h^{r_3} = g^{m_1 m_2} h^{r_1 m_2 + r_3}$ for the message $m_1 m_2$. To this purpose, she runs the following sigma protocol with Bob:

   - Alice selects random $m_1', m_2', r_1', r_2', r_3'$ all from $\mathbb{Z}_q$ and sends to Bob the commitments $a_1 = g^{m_1'} h^{r_1'}$, $a_2 = g^{m_2'} h^{r_2'}$ and $a_3 = c_1^{m_2'} h^{r_3'}$.
   - Bob sends a random $e$ back to Alice.
   - Alice sends the responses $f_{m_1} = m_1' + e m_1$, $f_{m_2} = m_2' + e m_2$, $f_{r_1} = r_1' + e r_1$, $f_{r_2} = r_2' + e r_2$, $f_{r_3} = r_3' + e r_3$ to Bob.

   (a) What equations should be verified by Bob in order to check this proof?

   (b) Show that this protocol is honest verifier zero-knowledge.

   (Note that we do not ask you to care about the soundness or the completeness of this protocol.)

4. We build a bit commitment scheme based on a hash function $\Pi = \langle \mathsf{Gen}, H \rangle$: $\mathsf{Gen}$ picks a random hash function index $s$. Then, $\mathsf{Com}_s(m)$ provides the commitment $c = H^s(b \| r)$ for a random $r$ that is as long as the output of $H^s$ and the opening $d = (r, b)$. $\mathsf{Open}$ is defined in the natural way. Show that this scheme is computationally binding if the hash function $\Pi$ is collision-resistant.