Université Libre de Bruxelles
Faculté des Sciences

January 22, 2020

INFO-F-405 Introduction to cryptography

This assessment has to be done without the use of personal notes, books or any other support. You may use a calculator (but it is not required). You have to justify and detail each of your answers. Indicate your name, first name and year of study on every answer sheet. The answers to the questions should be on different answers sheets (on sheet per question; the multiple-choice questions have to be together on one sheet).

Question 1

(3.8/20) Secret-key encryption The Electronic Codebook (ECB) mode is a flawed encryption mode on top of a block cipher.

a) What makes it an insecure mode?

b) What could an adversary make as queries and choice of (m_0, m_1) to win the IND-CPA game?

c) Describe a mode that turns a block cipher into a stream cipher. How can this mode securely handle the encryption of multiple messages using the same secret key?

Note: both ECB and IND-CPA are given in the portfolio.

Question 2

(3.8/20) Hashing Let $\mathcal{RO}(x)$ be a random oracle that outputs an infinite sequence of uniformly and independently distributed bits for each input $x \in \{0,1\}^*$. If the random oracle is queried twice with the same input, it always returns the same sequence. We denote with $\mathcal{RO}(x, \ell)$ the output of $\mathcal{RO}(x)$ truncated after the first ℓ output bits, so $\mathcal{RO}(x, \ell) \in \{0,1\}^\ell$.

a) *Preimage attack.* Given some value $y \in \{0,1\}^\ell$, an adversary would like to find a preimage, i.e., an input x such that $\mathcal{RO}(x, \ell) = y$. What is the probability of success after t random attempts?

b) *Multi-target preimage attack.* Given a set of m distinct values $\{y_1, \ldots, y_m\}$, with $y_i \in \{0,1\}^\ell$, an adversary would be happy to find a preimage of any one of these images. What is the probability of success after t random attempts?

Let $h(x)$ be a concrete extendable output function (XOF) that outputs a potentially infinite sequence of bits, and we similarly denote $h(x, \ell)$ its truncation to ℓ output bits. As of today, the only known attack against $h(x)$ has a probability of success $\epsilon(t) = t/2^{c/2}$, for some security parameter c, and where t is the time spent by the adversary expressed in the number of attempts. This means that, except for this probability $\epsilon(t)$, we can model $h(x)$ as a random oracle.

a) What is the range of output sizes ℓ and parameter values c such that $h(x)$ offers 128 bits of preimage resistance?

b) What is the range of output sizes ℓ and parameter values c such that $h(x)$ offers 128 bits of multi-target preimage resistance with $m = 2^{32}$?

You may use the approximation $\log(a + b) \approx \max(\log a, \log b)$.

Question 3

(3.8/20) Public-key signature with Schnorr Consider the Schnorr-type signature scheme based on elliptic curves in the portfolio.

a) What does the notation $X = [y]Z$ mean?

b) Please explain why the verification succeeds on a genuinly generated signature, and expand the computations annotated with the properties used.

c) The value k must be randomly generated and unique per signature. What happens if k is repeated?

d) To make k random and unique, the signer generates a random k upon the first signature, and then increments it ($k \leftarrow k + 1$) for each new signature. Is that secure? If so, please justify. If not, what is the problem and how can you fix it?

Question 4

(3.8/20) Practical question A company that installs and maintains an open-source operating system is creating an automatic update mechanism for its customers. Each computer connected to the internet can fetch an *update pack* containing the latest changes to be made to the operating system. The company would like to guarantee the integrity of these update packs so as to avoid the installation of malicious software on their customers' computers.

Please propose a concrete solution based on schemes such as encryption, either public-key or secret-key, authentication, authenticated encryption, signature and hashing. You also need to describe how the keys are generated and/or distributed.

Question 5

(4.8/20) Multiple choices on various subjects For each of the following sub-questions, the relative rating is: 0.8 points for a correct answer, 0.0 for a wrong answer and 0.3 for “I don't know”.

A. After how many attempts can one typically find a collision on a secure hash function with an output length of 384 bits?

(0) I don't know.

(1) 128

(2) 192

(3) 384

(4) 2^{128}

(5) 2^{192}

(6) 2^{384}

B. What happens if the one-time pad is incorrectly used and that two distinct plaintexts are encrypted with the same key?

(0) I don't know.

(1) The key is compromised.

(2) The two plaintexts are revealed.

(3) The difference between the two plaintexts is revealed.

(4) The authenticity of the plaintext is compromised.

C. What is the relationship between the discrete logarithm (DL) and the Diffie-Hellman (DH) problems?

(0) I don't know.

(1) An adversary who can solve the DL problem can also solve the DH problem, but not necessarily vice-versa.

(2) An adversary who can solve the DH problem can also solve the DL problem, but not necessarily vice-versa.

(3) Both problems are equivalent.

D. How does hybrid encryption work?

(0) I don't know.

(1) Alice sends a secret key to Bob encrypted with his public key, and she encrypts the plaintext with the secret key.

(2) Alice sends a public key to Bob encrypted with their secret key, and she encrypts the plaintext with his public key.

(3) Alice sends a public key to Bob encrypted with her private key, and she encrypts the plaintext with her public key.

(4) Alice sends a private key to Bob encrypted with his public key, and she encrypts the plaintext with his public key.

E. Let E be an elliptic curve over $\mathrm{GF}(p)$ for a given prime number p and let $\#E$ be the number of points on E. Which statement is most plausible? (The symbol $\sim$ means “is of the order of”.)

(0) I don’t know.

(1) $p \sim 2^{512}$, $\#E \sim 2^{256}$ and E offers a security of ≈ 128 bits.

(2) $p \sim 2^{256}$, $\#E \sim 2^{128}$ and E offers a security of ≈ 128 bits.

(3) $p \sim 2^{256}$, $\#E \sim 2^{256}$ and E offers a security of ≈ 128 bits.

F. Assume an adversary performs an exhaustive key search on a huge network of 10^9 computers, each capable of testing 10^9 keys per second. After about how much time will a 128-bit key typically be found?

(0) I don’t know.

(1) A few seconds.

(2) A few days.

(3) A few years.

(4) A few centuries.

(5) A few times the age of the universe.

Portfolio

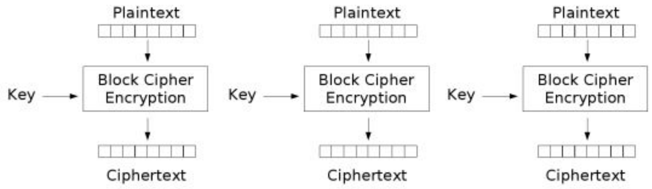
Electronic Codebook (ECB)

Input: secret key $k \in \{0,1\}^m$, plaintext $p \in \{0,1\}^*$, output: ciphertext $c \in \{0,1\}^*$

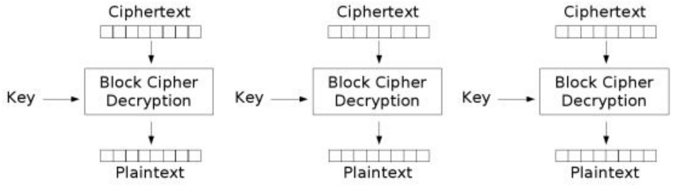
- Pad p with 10^* to reach a multiple of the block size n
- Cut $p \| 10^*$ into blocks p_i of size n
- Process each block independently through E_k

$$c_i = E_k(p_i)$$

And similarly with E_k^{-1} for decryption.



Electronic Codebook (ECB) mode encryption



Electronic Codebook (ECB) mode decryption

Indistinguishability under chosen plaintexts (IND-CPA)

A scheme $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is **IND-CPA-secure** if no adversary can win the following game for more than a negligible advantage.

- Challenger generates a key (pair) $k \leftarrow \mathrm{Gen}()$
- Adversary queries Enc_k with plaintexts of his choice
- Adversary chooses two plaintexts $m_0, m_1 \in M$ with $|m_0| = |m_1|$
- Challenger randomly chooses $b \leftarrow_R \{0,1\}$, encrypts m_b and sends $c = \mathrm{Enc}_k(m_b)$ to the adversary
- Adversary queries Enc_k with plaintexts of his choice
- Adversary guesses the index b' of the plaintext was encrypted
- Adversary wins if $b' = b$

Schnorr-type signature scheme on elliptic curve

Key pair on curve E and base point G of order q

- Private key $a \in \mathbb{Z}_q$
- Public key $A = [a]G$

Signature of message $m \in \{0,1\}^*$ by Alice with her private key a:

- Randomly choose k in $\mathbb{Z}_q$
- Compute $R = [k]G$
- Compute $e = \text{hash}(R \| A \| m)$
- Compute $s = k + ea \bmod q$
- Send (R, s) along with m

Verification of signature (R, s) on m with Alice's public key A:

- Check $[s]G \overset{?}{=} R + [\text{hash}(R \| A \| m)]A$