Université Libre de Bruxelles
Faculté des Sciences

August 25, 2020

INFO-F-405 Introduction to cryptography

This assessment may be done with the use of personal notes, slides, books, web pages, etc. However, the assessment is strictly personal and you are not allowed to communicate with other students or other people during this assessment.

Question 1

(5.6/20) Public-key cryptography In the following, we describe a key agreement protocol, based on ephemeral Diffie-Hellman and on elliptic curves, that Alice and Bob use to obtain a common secret key k_{AB}. However, we voluntarily added some mistakes (including omissions) to its definition. Some of them are functional, that is, they prevent the protocol from working correctly, while others introduce security flaws.

Please *list all the mistakes* you find, and for each, *justify* why it is incorrect and *propose a correction.*

Let $\mathcal{E}$ be an elliptic curve over $\mathbb{Z}_p$ (for some prime p) that two parties, called Alice and Bob, agreed upon. They also agreed on a base point $G \in \mathcal{E}$ with prime order q, i.e., $[q]G$ is the neutral element.

Note that we describe the protocol only from Alice's point of view. The protocol from Bob's point of view is exactly the same, but with reverted roles and variables (i.e., $a, A, e, E, n_{\mathrm{A}}$ become $b, B, f, F, n_{\mathrm{B}}$, respectively, and vice-versa). Also, note that the uppercase letters refer to points on $\mathcal{E}$, while lowercase letters are integers.

Alice's key generation:

1. Alice secretly generates her secret key a randomly and uniformly from $[1 \ldots p-1]$.
2. Alice computes her public key $A = [a]G$, and publishes it via some PKI process. (This aspect is out of scope of this question. In the sequel, we assume that all the public keys are correctly authenticated and can be trusted.)
3. Alice secretly generates her secret value n_{A} randomly and uniformly from $[1 \ldots p-1]$.

Key agreement protocol, from Alice's point of view

1. Alice generates e randomly and uniformly from $[1 \ldots p-1]$.
2. Alice computes $E = [e]G$.
3. Alice signs E by calling $\mathrm{Sign}_a(E)$.
4. Alice sends both E and $\mathrm{Sign}_a(E)$ to Bob.
5. Alice receives $F = [f]G$ from Bob.

6. Alice computes $K_{\mathrm{AB}} = [e]F$.

7. Alice computes the common secret key $k_{\mathrm{AB}} = \mathrm{hash}(K_{\mathrm{AB}} \| a \| e)$.

<u>Procedure $\mathrm{Sign}_a(m)$</u>

1. Hash m: $h = \mathrm{hash}(m)$.
2. Compute $R = [n_{\mathrm{A}}]G$ and set r to the x-coordinate of R modulo p.
3. If $r = 0$, randomly generate a new n_{A} and restart from line 2.
4. Compute $s = n_{\mathrm{A}}^{-1}(h + ar) \bmod p$.
5. Return (R, s).

<u>Procedure $\mathrm{VerifySignature}_a(m, R, s)$</u>

1. Check that $R \in \mathcal{E}$, otherwise the signature is invalid.
2. Hash m: $h = \mathrm{hash}(m)$.
3. Set r to the x-coordinate of R modulo p.
4. Compute $A = [a]G$.
5. Compute $[r]A - [s]R$ and $[h]G$.
6. The signature is valid iff $[r]A - [s]R = [h]G$.

Question 2

(3.2/20) Hashing Let $f : \{0,1\}^* \to \{0,1\}^n$ be a one-way collision-resistant compressing function and $\mathsf{E}_k : \{0,1\}^n \to \{0,1\}^n$ a block cipher with key $k \in \{0,1\}^n$. For a $2n$-bit integer x, we write x_H to denote the n first (most significant) bits of x and x_L to denote the n last (least significant) bits of x such that $x = x_H || x_L$. We construct a compressing function $g : \{0,1\}^* \to \{0,1\}^{2n}$ as follows:

$$g(x) = \begin{cases} \mathsf{E}_{x_L}(x_H) || x_L & \text{if } |x| = 2n \\ f(x) || f(x) & \text{otherwise.} \end{cases}$$

where $|x|$ denote the bit-size of x.

Is $g(x)$ a collision resistant function? What about one-wayness?

Be careful to formally motivate your answer !

Question 3

(5.6/20) Practical application You have been appointed as the new project manager of the COVID-19 tracing team. The goal of your project is to prevent the spread of the virus by letting people know when they have been in contact with an infected person. A trivial solution would be to force every person receiving a positive result for their test to publicly announce where they have been and who they met in the last 14 days. Naturally, this would be unacceptable as it would be a huge infringement of privacy. As an ethic technology enthusiast, you realize that cryptography can be used to construct a less invasive solution, more respectful of privacy.
We ask you to depict such a solution by giving a high-level explanation of your idea followed by a formal description of the cryptographic primitives used during the communications. In addition to that, we ask you to explain what are the different privacy and security threats that you identified in your analysis of the situation and how your solution is actually preventing them. The question is rather open, but we will make the following assumptions to simply the situation:

- Every person possess a smartphone that can be loaded with code from a trusted source.
- Those smartphones can communicate over the Internet as well as in short-range in a peer-to-peer fashion.
- Tests are performed by honest medical doctors who have access to the Internet and can retrieve information from the phone of their patient.
- A global database is available online.

Question 4

(5.6/20) Inside symmetric primitives Below are the specifications of a block cipher under development, called ABCD. We give the specifications of the evaluation of the block cipher in the forward direction, ask you some questions about it, then we ask you to write the specifications of the inverse block cipher.

ABCD is a 256-bit block cipher with a variable-length key. The authors were not very much inspired and gave it its name because the 256-bit input block and running state are represented as 4 words (or rows) of 64 bits each, denoted a, b, c and d. We denote the individual bits in a rows using subscripts, e.g., row a is composed of the bits $a_0, a_1, \ldots, a_{63}$. The four bits (a_i, b_i, c_i, d_i) at the same position in each row is called a column. In other words, the state can be viewed as a grid of 4 rows by 64 columns.

The evaluation of the block cipher consists in 13 rounds. We do not give the specifications of the key schedule, but we assume that, from the input secret key K, one can derive 14 round keys $K_0, K_1, \ldots, K_{13}$ of 256 bits each. The evaluation of block cipher goes as follows:

```
AddRoundKey($K_0$)
for each round $i = 1$ to 13 do
  ShakeColumns
  PreShiftRows
  StirColumns
  PostShiftRows
```

```
    AddRoundKey($K_i$)
  end for
```

We now describe the five step mappings AddRoundKey, ShakeColumns, PreShiftRows, StirColumns and PostShiftRows, all specified at the bit level, or more formally, in the Galois field $\mathrm{GF}(2) = \{0, 1\}$. In this field, $x + y$ (resp. xy) denotes the modulo-2 addition (resp. multiplication) of $x, y \in \mathrm{GF}(2)$. We use the operator $\oplus$ to express the component-wise addition of vectors of elements in GF(2), such as rows, columns or states.

Definition of AddRoundKey(K_i)

```
  $(a, b, c, d) \leftarrow (a, b, c, d) \oplus K_i$
```

Definition of ShakeColumns

```
  for each column $i = 0$ to 63 do
    $p \leftarrow a_i + b_i + c_i + d_i$
    $a_i \leftarrow a_i + p$
    $b_i \leftarrow b_i + p$
    $c_i \leftarrow c_i + p$
    $d_i \leftarrow d_i + p$
  end for
```

Definition of PreShiftRows

```
  $a \leftarrow a$
  $b \leftarrow \mathrm{rot}^1(b)$
  $c \leftarrow \mathrm{rot}^3(c)$
  $d \leftarrow \mathrm{rot}^9(d)$
```

For a row x, $\mathrm{rot}(x)$ denotes the operation that consists in cyclically shifting all the bits by one position, i.e.,

$$(x_0, x_1, x_2, \ldots, x_{63}) \to (x_1, x_2, \ldots, x_{63}, x_0).$$

Definition of StirColumns

```
  for each column $i = 0$ to 63 do
    $a_i \leftarrow a_i + b_i c_i$
    $b_i \leftarrow b_i + c_i d_i$
    $c_i \leftarrow c_i + d_i a_i$
    $d_i \leftarrow d_i + a_i b_i$
  end for
```

Definition of PostShiftRows

```
  $a \leftarrow a$
  $b \leftarrow \mathrm{rot}^1(b)$
  $c \leftarrow \mathrm{rot}^5(c)$
  $d \leftarrow \mathrm{rot}^{25}(d)$
```

Sub-question A: Classify each step mapping as linear or non-linear. As a reminder, a mapping λ is linear iff $\forall x, y : \lambda(x \oplus y) = \lambda(x) \oplus \lambda(y)$. For linear mappings, please provide a short justification (one line). For non-linear mappings, please provide an example of x, y that violates the definition of linearity.

Sub-question B: Point out the step mapping(s) that provide diffusion, with a short justification. As a reminder, a step mapping provides diffusion if at least one input bit influences more than one output bit.

Sub-question C: For the step mapping ShakeColumns, please explain what happens at the output when:

- one flips 1 bit at the input;
- one flips 2 bits of the same column at the input;
- one flips 2 bits of different columns at the input.

Sub-question D: Write the specifications of the inverse of ABCD.