Cryptography and Security — Final Exam
Solution

Serge Vaudenay

29.1.2026

- duration: 3h00
- no documents allowed, except one 2-sided sheet of handwritten notes
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **not** answer any technical question during the exam
- readability and style of writing will be part of the grade
- answers should not be written with a pencil

The exam grade follows a linear scale in which each question has the same weight.

1 Mersenne Cryptosystem

The following exercise is inspired from A New Public-Key Cryptosystem via Mersenne Numbers *by Aggarwal, Joux, Prakash, and Santha, published in proceedings of CRYPTO 2018.*

A Mersenne number is an integer of the form $p = 2^n - 1$. When it is prime, it is called a Mersenne prime. There exists a few known large Mersenne primes such as $p = 2^{756\,839} - 1$.

In this exercise, we fix n such that $p = 2^n - 1$ is prime. Given an integer x, we denote by $\mathsf{HW}(x)$ the Hamming weight of the binary representation of $x \bmod p$ (i.e. the number of bits which are equal to 1 once x has been reduced modulo p).

Q.1 For any list $(s_1, \ldots, s_k)$ of non-negative integers (not necessarily bounded by n), prove that $\mathsf{HW}(2^{s_1} + \cdots + 2^{s_k}) \leq k$.

Since $p = 2^n - 1$, we have $2^n \equiv 1 \pmod p$ so $2^i \equiv 2^{i \bmod n} \pmod p$. Thus, we assume without loss of generality that every s_i is in the range $\{0, \ldots, n-1\}$.
If there is no repetition in the list, $\left(\sum_{i=1}^k 2^{s_i}\right) \bmod p$ has as a binary representation a bit string with 1 set at positions corresponding to every s_i and 0 set at positions elsewhere. So, the result is quite clear. What is tricky is to prove the result with repetitions.
We prove the result by induction on k.
For $k = 0$, this is quite clear.
Assuming the result is proven for $k = \kappa - 1$, $\kappa > 0$, we prove it for $k = \kappa$ as follows.
If there is no repetition, the result is proven with the previous argument.
Otherwise, say $s_i = s_j$ for some specific i and j. Since the result does not depend on the order of the indices, let us assume without loss of generality that $i = k-1$ and $j = k$. We have $2^{s_i} + 2^{s_j} = 2^{s_j+1}$. So, the result for $(s_1, \ldots, s_k)$ is equivalent to the result for $(s_1, \ldots, s_{k-2}, s_k + 1)$ which has length $\kappa - 1$, so $\mathsf{HW}(2^{s_1} + \cdots + 2^{s_k}) \leq \kappa - 1$, due to the induction assumption. Since $\kappa - 1 \leq \kappa$, this proves the induction.

Q.2 For any integer A and B, prove that $\mathsf{HW}(A+B) \leq \mathsf{HW}(A) + \mathsf{HW}(B)$ and $\mathsf{HW}(AB) \leq \mathsf{HW}(A) \times \mathsf{HW}(B)$.

We write $(a_1, \ldots, a_\alpha)$ the list of bit positions (without repetition) of $A \bmod p$ which have a bit set to 1. We have $A \bmod p = \sum_{i=1}^{\alpha} 2^{a_i}$ and $\mathsf{HW}(A) = \alpha$. Similarly, $B \bmod p = \sum_{i=1}^{\beta} 2^{b_i}$ and $\mathsf{HW}(B) = \beta$.
We have $A + B \equiv \sum_{i=1}^{\alpha} 2^{a_i} + \sum_{i=1}^{\beta} 2^{b_i} \pmod p$ which corresponds to the list $(a_1, \ldots, a_\alpha, b_1, \ldots, b_\beta)$ of length $\alpha + \beta$. Due to the previous question, $\mathsf{HW}(A+B) \leq \alpha + \beta = \mathsf{HW}(A) + \mathsf{HW}(B)$.
Similarly, $AB \equiv \sum_{i,j} 2^{a_i + b_j} \pmod p$ which corresponds to the list $(a_i + b_j)_{i,j}$ of size $\alpha\beta$. Hence, $\mathsf{HW}(AB) \leq \alpha\beta = \mathsf{HW}(A) \times \mathsf{HW}(B)$.

Q.3 For $X \in \mathbf{Z}_p$ uniformly distributed and a constant integer s, give the distribution of $\mathsf{HW}(X \oplus Y)$ where $Y = (X + 2^s) \bmod p$. (Here, $\oplus$ is the bitwise exclusive or of the modulo p binary representation.) Namely, prove that $\Pr[\mathsf{HW}(X \oplus Y) = i] = \frac{2^{n-i}}{p}$ for $i = 1, \ldots, n$.

HINT: reduce to the $s = 0$ case.

We observe that the $Z \mapsto (Z2^{-s}) \bmod p$ function is actually a circular rotation of the bits of $Z \bmod p$ by s positions to the right. Hence,

$$\mathsf{HW}(X \oplus Y) = \mathsf{HW}(((X2^{-s}) \bmod p) \oplus ((Y2^{-s}) \bmod p))$$

By setting $X' = (X2^{-s}) \bmod p$ and $Y' = (Y2^{-s}) \bmod p = (X'+1) \bmod p$, we reduce to the $s = 0$ case.
Adding 1 to X affects the i least significant bits if and only if the binary representation of X ends by $011\cdots1$ with $i-1$ tailing 1's. This can only be for $i = 1, \ldots, n$. The number of such X in $\mathbf{Z}_p$ is 2^{n-i}. Hence, $\Pr[\mathsf{HW}(X \oplus Y) = i] = \frac{2^{n-i}}{p} \approx 2^{-i}$ for $i = 1, \ldots, n$.
There was a little mistake in this question: if $X = 01\cdots1$, then $Y = 10\cdots0$ and $X \oplus Y = 1\cdots1$ but $\mathsf{HW}(1\cdots1) = 0$ by definition, due to the modulo p reduction. The result is correct for $\Pr[d_H(X,Y) = i]$ with $d_H(X,Y)$ being the Hamming distance between the modular reductions of X and Y, but when the $\oplus$ is done before the reduction, the probability should restrict to $i = 1, \ldots, n-1$.

Q.4 For $X \in \mathbf{Z}_p$ uniformly distributed and a constant δ such that $\mathsf{HW}(\delta) = k$, we let $Y = (X + \delta) \bmod p$. Prove that $\Pr[\mathsf{HW}(X \oplus Y) \geq i] \leq k2^{2-\frac{i}{k}}$.

HINT: set $\delta = \sum_{i=1}^{k} 2^{s_i}$, $X_0 = X$, and $X_i = (X_{i-1} + 2^{s_i}) \bmod p$.

Clearly, every X_i is uniformly distributed in $\mathbf{Z}_p$, and $X_k = Y$. We have $\mathsf{HW}(X \oplus Y) \leq \sum_{i=1}^{k} \mathsf{HW}(X_{i-1} \oplus X_i)$, by triangular inequality. If $\mathsf{HW}(X \oplus Y) \geq i$, it must be the case that there is one index i' such that $\mathsf{HW}(X_{i'-1} \oplus X_{i'}) \geq \frac{i}{k}$. Thanks to the previous question, we know that

$$\Pr\left[\mathsf{HW}(X_{i'-1} \oplus X_{i'}) \geq \frac{i}{k}\right] = \sum_{j=\lceil \frac{i}{k} \rceil}^{n} \frac{2^{n-j}}{p} \leq \sum_{j=\lceil \frac{i}{k} \rceil}^{+\infty} \frac{2^{n-j}}{p} = 2\frac{2^{n-\lceil \frac{i}{k} \rceil}}{p} \leq 2\frac{2^{n-\frac{i}{k}}}{p}$$

So,

$$\Pr[\mathsf{HW}(X \oplus Y) \geq i] \leq \sum_{i'=1}^{k} \Pr\left[\mathsf{HW}(X_{i'-1} \oplus X_{i'}) \geq \frac{i}{k}\right] \leq k \times 2\frac{2^{n-\frac{i}{k}}}{p} \leq k2^{2-\frac{i}{k}}$$

where we used $2^n \leq 2p$.
We note that there is a better bound in the original paper, but it is harder to obtain.

Q.5 Given a security parameter λ, we define some parameters h, n, p such that $h = \lambda$, $10h^2 \leq n \leq 16h^2$, $p = 2^n - 1$, and p prime. Using these parameters, a key generation algorithm $\mathsf{Gen}(h, n, p)$ first picks at random $F, G, R \in \mathbf{Z}_p$ such that $\mathsf{HW}(F) = \mathsf{HW}(G) = h$. Then, it sets $T = (FR + G) \bmod p$, $\mathsf{pk} = (R, T)$, and $\mathsf{sk} = F$. The output is pk and sk. Moving ahead, we define a cryptosystem with Gen as a key pair generator. We consider the problem of winning in the following game (where the winning condition is missing):

Input: h, p
1: pick $(F, G, R) \in \mathbf{Z}_p^3$ satisfying $\mathsf{HW}(F) = \mathsf{HW}(G) = h$ uniformly
2: $T \leftarrow (FR + G) \bmod p$
3: $F' \leftarrow \mathcal{A}(h, p, R, T)$
4: **return** 1_{win}

Write down the game defining security against key recovery under chosen plaintext attacks and define the winning condition in the above game so that the two games become equivalent.

No matter what the encryption and decryption algorithms are, the adversary has input pk and no oracle. (Chosen plaintext encryption is done by the adversary using the public key.)

Input*: h, p*
1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(h, p)$
2: $\mathsf{sk}' \leftarrow \mathcal{A}(h, p, \mathsf{pk})$
3: ***return*** $1_{\mathsf{sk}'=\mathsf{sk}}$

The winning condition in the proposed game is $F = F'$. It makes the two games identical but with notations.

Q.6 We assume an efficient encoding algorithm $\mathcal{E}$ mapping a message m in a message domain to $\mathcal{E}(m) \in \mathbf{Z}_p$ and an efficient decoding algorithm $\mathcal{D}$ such that $\mathcal{D}(\mathcal{E}(m) \oplus e) = m$ for any message m and any $e \in \mathbf{Z}_p$ satisfying $\mathsf{HW}(e) \leq \tau$, for a given threshold τ. We define the encryption of m as follows: we pick A, B_1, B_2 in $\mathbf{Z}_p$ satisfying $\mathsf{HW}(A) = \mathsf{HW}(B_1) = \mathsf{HW}(B_2) = h$ uniformly, then $C_1 = (AR + B_1) \bmod p$ and $C_2 = ((AT + B_2) \bmod p) \oplus \mathcal{E}(m)$,

and output $\mathsf{ct} = (C_1, C_2)$. The decryption of ct is $m' = \mathcal{D}(((FC_1) \bmod p) \oplus C_2)$. Prove that the cryptosystem is correct but for some probability up to ε to upper bound based on h, p, and τ.

(We omit the modulo p reductions in this question for more readability.) If the encryption was honestly done from a honestly generated key, we have $C_1 = AR + B_1$, $y = AT + B_2$, and $C_2 = y \oplus \mathcal{E}(m)$. Furthermore, $T = FR + G$ in key generation. Hence, the decryption computes $x = FC_1 = F(AR + B_1) = AFR + FB_1$. Furthermore, $y = A(FR+G)+B_2 = AFR+AG+B_2$. So, if we define $X = AFR$, $\delta_1 = FB_1$, and $\delta_2 = AG + B_2$, we have $\mathsf{HW}(x \oplus y) \leq \mathsf{HW}(X \oplus (X + \delta_1)) + \mathsf{HW}(X \oplus (X + \delta_2))$, by triangular inequality.
Due to the properties of the Mersenne numbers, we have $\mathsf{HW}(\delta_1) \leq h^2$ and $\mathsf{HW}(\delta_2) \leq h^2 + h$. Due to a previous question with $k = \frac{h^2+h}{2}$, we have

$$\Pr[\mathsf{HW}(x \oplus y) \geq \tau] \leq 2k2^{2-\frac{\tau}{k}} = k2^{3-\frac{\tau}{k}} = (h^2 + h)2^{2-\frac{2\tau}{h^2+h}}$$

We can thus set $\varepsilon = (h^2+h)2^{2-\frac{2\tau}{h^2+h}}$. Except with probability ε, we have $\mathsf{HW}(x \oplus y) < \tau$ so decoding $x \oplus C_2$ is just decoding $x \oplus y \oplus \mathcal{E}(m)$, which is close enough to $\mathcal{E}(m)$. Hence, it gives m.

2 Nonce Repetition in ML-DSA

We recall the ML-DSA signature algorithm as seen in class. We use

$$q = 8\,380\,417 = 2^{13} \times 3 \times 11 \times 31 + 1$$

$R_q = \mathbf{Z}_q[X]/(X^{256}+1)$, and some parameters k and ℓ (say for example $k = \ell = 4$). We use a hash function H with range in the set of small elements of R_q. We also define a function round from $\mathbf{Z}_q$ to $\mathbf{Z}_q$ (that we extend to a function from R_q^i to R_q^i by applying to each coefficient) which rounds a modulo-q residue to the nearest one which is ending by ℓ zero bits.

- To generate a key pair, we pick $A \in R_q^{k\times\ell}$ uniformly and $s_1 \in R_q^\ell$ small. We set $t_1 = \mathsf{round}(As_1)$. Finally, $\mathsf{sk} = s_1$ and $\mathsf{pk} = (A, t_1)$.
- To sign a message M, we compute $\mu = H(0\|M)$, we pick a nonce $y \in R_q^\ell$ small, we set $w = \mathsf{round}(Ay)$, $c = H(\mu, w)$, $z = y + cs_1$, and the signature is $\sigma = (c, z)$.
- To verify a signature σ for M, we check that z is small. We compute μ and $w_\approx = Az - ct_1$. We check that $c = H(\mu, \mathsf{round}(w_\approx))$.

Q.1 Given two different signed messages which are using the same nonce, show how to do a key recovery attack.

Given two signed messages (M, c, z) and (M', c', z') using the same $y' = y$, we have $z = y + cs_1$ and $z' = y + c's_1$ so $s_1 = \frac{z'-z}{c'-c}$ which yields the secret key.

Q.2 Given a collection of n signed messages in which two are using the same nonce, give an algorithm to identify those two signed messages and analyze the complexity.

Let (M_i, c_i, z_i) be the ith signed message, y_i be the used nonce, and μ_i and w_i being the intermediate computations. We have that $y_i = y_j$ implies $w_i = w_j$ but c_i and c_j are the result of hashing the message with it so the nonce repetition is not directly visible from the signature.
What we can do is to apply the previous attack for every pair (i, j) then to check if the recovered s_1 satisfies $t_1 = \mathsf{round}(As_1)$. The complexity is equivalent to $\frac{n^2}{2}$ operations $s_1 \mapsto \mathsf{round}(As_1)$.

Q.3 What is the improvement about nonce-misuse issues compared to DSA or ECDSA?

In DSA or ECDSA, one element is a deterministic function of the nonce so we can detect a nonce reuse in quasi linear time using a dictionary. The complexity to find a repetition in a list of n signed messages is thus much lower.

3 Machine-Readable Travel Documents

We recall the principle of machine-readable travel documents (MRTD). If a holder shows the opened MRTD to a reader, the reader can optically read a machine-readable zone (MRZ). It contains some information which is used as a password. It consists of: a serial number (8 upper-case alphanumerical characters), a date of birth, an expiration date. Given this information, the reader and the NFC chip can run a password authenticated key exchange (PAKE) and open a secure communication channel. Then, the chip provides the reader with the mandatory elements: a digital copy of the MRZ (called DG1), a picture to be used as a model for facial recognition (called DG2), an element SOD. The SOD includes the list of the hash of each DGi which is present in the chip and requires passive authentication, the signature of this list by the issuer, and the certificate of the issuer. The MRZ contains other information such as the name of the holder, their nationality and their gender.

Q.1 What is the difference between the digital image in DG2 and a regular digital picture?

The image in DG2 is optimized for automated facial recognition. The photoshooting is made with a good equipment (camera and flash), optimized enlightement, no color, and drastic requirements (such as no glasses, no hat, no smile). Compared to Instagram pictures, DG2 can recognize a face much better.

Q.2 Compared to traditional travel documents, explain what the SOD leaks.

SOD leaks digital evidence. A digital signature is an undeniable evidence, contrarily to a photocopy of a traditional passport, because photocopies can easily be forged. This evidence can be used to prove the official name, gender, date-of-birth, and nationality of the person corresponding to a digital picture. Once published by an adversary, it cannot be denied.
Besides, SOD also leaks the hash of DGi which are not readable without terminal authentication. This can contain more private objects which are not necessarily printed in the passport.

Q.3 Estimate the entropy (in bits) of the password.

Given that a passport is typically valid for 10 years, the expiration date has only $3650 \approx 2^{12}$ *possibilities. Assuming we can reliably estimate the age of a person with a* ± 5*years margin, the entropy of the date of birth is also of* 12 *bits. The alphabet for the serial number has* $26 + 10$ *characters so we have up to* $36^8 \approx 2^{41}$*. The total entropy is thus bounded by* 65 *bits.*
However, with a serial number having 2 digits, then 2 letters, then 4 digits, the entropy of the serial number becomes 29 *so the total entropy is now of* 43 *bits.*
If we already know the person and if their date of birth can be easily found based on public information such as Wikipedia, the entropy estimate drops by 12 bits.

Q.4 Compare the security of the password with the one of a regular password chosen by a human user. What is the risk of a disclosed MRTD password?

Compared to the entropy of a human-chosen password, the entropy is not too bad. However, the information based on which the MRZ password is made is often used and may be collected by many organizations and treated without care. For instance, people often need to show their passport at the check-in counter in a hotel and the hotel typically collect all information. They can be hacked and the information ends up on the darknet. The exposure of human-chosen passwords is normally much lower. If the MRTD password is known, we can make sensors to detect the presence of this MRTD hence track a person by their hidden passport.