Cryptography and Security — Final Exam

Serge Vaudenay

14.1.2016

- duration: 3h
- 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

1 Attack on DL-Based Signature Schemes

In what follows, we consider a cyclic group of order q generated by some element g. We let $\langle g \rangle$ denote this group and we take multiplicative notations. We let 1 denote the neutral element. We assume that comparing and multiplying two group elements is easy and that inverting an element is easy. We assume that the discrete logarithm problem is hard in this group. In particular, we assume that $q > 2^{160}$. We further assume that we have a hash function G mapping an arbitrary group element to a $\mathbf{Z}_q$ element and a hash function H mapping an arbitrary bitstring to a $\mathbf{Z}_q$ element.

Q.1 We consider a digital signature scheme (inspired by DSA) in which the key generation and the signature algorithm work as follows:

Key generation:

```
1: pick $x \in \mathbf{Z}_q$ with uniform distribution
2: compute $y = g^x$
3: set the secret key to $x$ and the public key to $y$
```

Sign m **using key** x:

```
1: pick $k \in \{1, 2, \ldots, 2^{128}\}$ with uniform distribution
2: compute $r = G(g^k)$
3: compute $s = \frac{H(m)+xr}{k} \bmod q$
4: set the signature to $(r, s)$
```

Verify signature (r, s) **for** m **using key** y:

```
1: check that $G\left(g^{\frac{H(m)}{s} \bmod q} y^{\frac{r}{s} \bmod q}\right) = r$
```

Prove that under a honest execution, a signature is always correct.

Q.2 Assume that an adversary collects many signed messages (m_i, r_i, s_i) for $i = 1, 2, \ldots, n$. If $r_i = r_j$ for $i < j$, show that the adversary can easily make a key recovery attack. How large must n be for this to happen?

HINT: first prove by an informal probability estimate that $r_i = r_j$ is most likely due to $k_i = k_j$.

Q.3 To defeat the previous attack, our usual crypto apprentice designs the following signature scheme:

Key generation:

```
pick $x_1 \in \mathbf{Z}_q$ with uniform distribution
pick $x_2 \in \mathbf{Z}_q$ with uniform distribution
compute $y_1 = g^{x_1}$ and $y_2 = g^{x_2}$
set the secret key to $(x_1, x_2)$ and the public key to $(y_1, y_2)$
```

Sign m using key (x_1, x_2):

```
pick $k \in \{1, 2, \ldots, 2^{128}\}$ with uniform distribution
compute $r_1 = G_1(g^k)$ and $r_2 = G_2(g^k)$
compute $s = \frac{H(m)+x_1r_1+x_2r_2}{k} \bmod q$
set the signature to $(r_1, r_2, s)$
```

where we now use two independent hash functions G_1 and G_2 to hash group elements onto $\mathbf{Z}_q$.

Propose a verification algorithm and prove that it works.

Q.4 The idea of the crypto apprentice is that to adapt the attack of Q.2 to this new scheme, one needs to find i, j, ℓ such that $i < j < \ell$ and $k_i = k_j = k_\ell$. With appropriate approximations, prove that we need $n \approx 2^{86}$ to have good chances of such i, j, ℓ to exist and conclude that this attack has a too high complexity.

HINT: approximate $\log \Pr[\textsf{no 3-collision}]$.

Q.5 Ignore the idea with 3-collisions and prove that two regular 2-collisions would suffice to break the new scheme. Say how large n should be for this better attack to work.

NOTE: we do not require a formula to give x_1 and x_2.

Q.6 Upset, the crypto apprentice decides to avoid collisions by using a counter in the following scheme:

Key generation:

```
pick $x \in \mathbf{Z}_q$ with uniform distribution
compute $y = g^x$
set the secret key to $x$ and the public key to $y$
set the counter $k$ to a random number
set the $e$ register to $g^k$
```

Sign m using key x:

```
increment the counter $k$
set $e$ to $eg$
compute $r = G(e)$
compute $s = \frac{H(m)+xr}{k} \bmod q$
set the signature to $(r, s)$
```

Design a key-recovery attack for this scheme using two signatures.

Q.7 What if we now use the following scheme?

Key generation:

```
pick $x \in \mathbf{Z}_q$ with uniform distribution
compute $y = g^x$
set the secret key to $x$ and the public key to $y$
set the counter $k$ to a random number
pick $\mathsf{inc} \in \mathbf{Z}_q^*$ with uniform distribution
set the $e$ register to $g^k$
set the $e'$ register to $g^{\mathsf{inc}}$
```

Sign m using key x:

```
set $k$ to $k + \mathsf{inc}$
set $e$ to $ee'$
compute $r = G(e)$
compute $s = \frac{H(m)+xr}{k} \bmod q$
set the signature to $(r, s)$
```

2 RSA in an Extension Ring

Q.1 Let p be a prime number such that $p \bmod 4 = 3$. We consider the polynomial $x^2 + 1$ in the ring $\mathbf{Z}_p[x]$ of polynomials in the indeterminate x, with coefficients in $\mathbf{Z}_p$. Prove that $x^2 + 1$ is irreducible.

Q.2 Let p be a prime number such that $p \bmod 4 = 3$. We consider the set $K = \mathbf{Z}_p[x]/(x^2 + 1)$ of all polynomials over $\mathbf{Z}_p$ taken modulo $x^2 + 1$. This defines the addition and the multiplication over K. (This is just the regular addition and multiplication of polynomials reduced modulo $x^2 + 1$ and modulo p.) Give the cardinality of K and say what type of algebraic structure it has. Justify your answer.

Q.3 Let p and q be two different prime numbers such that $p \bmod 4 = q \bmod 4 = 3$. Let $n = pq$. Let $R = \mathbf{Z}_n[x]/(x^2 + 1)$ be the set of all polynomials over $\mathbf{Z}_n$ taken modulo $x^2 + 1$. We want to construct an RSA-like cryptosystem over R.
Prove that there are exactly $\phi = (p^2 - 1)(q^2 - 1)$ invertible elements in R.
HINT: either count or think Chinese.

Q.4 Under the same hypothesis as in Q.3, we want to encrypt an element $m \in R$ by computing m^e and to decrypt by raising to the power d. How to set e and d for the decryption to work correctly? Justify your answer.

Q.5 In the context of Q.4, can we take $e = 3$? Justify your answer.

Q.6 By selecting the last decimal digit of p to be equal to 7 and the last decimal digit of q to be equal to 3, prove that we can always use $e = 5$ in the previous construction.

Q.7 Is there any advantage of this cryptosystem compared to RSA? Explain why.
HINT: compare the security with respect to modulus size, key and message lengths, and complexities.

3 On Securing Biometric Passports

A biometric passport is an identity document with a contactless chip. Reading the digital identity works like this:

1: The reader first reads the low-entropy password w which is printed inside the passport.
2: The reader sends a standard RFID broadcast signal and the chip responds.
3: The chip requests to go through a password-based key agreement. The password w is the input of the protocol on the reader side. On the chip side, there is a long-term public/secret key pair pk/sk and w. (sk is stored in the chip but is not accessible to the reader.) At the end of the protocol, the output on both sides is a symmetric key K.
4: The reader and the chip communicate securely by using this key K.
5: Through this secure communication, the reader can retrieve some files containing the identity information ID, a biometric reference template bio, the public key pk again, and a signature σ from the issuing country that $(\mathsf{ID}, \mathsf{bio}, \mathsf{pk})$ is correct.
6: The reader extracts from ID the field $\mathsf{country}$ indicating the issuing country. It is assumed that the reader has previously got in a secure way the root certificate C_{country} from the issuing country so that he can verify σ.

Then, the reader has obtained $(\mathsf{ID}, \mathsf{bio})$ which can then be used to identify the person.

We further describe BAC, the original password-based key agreement protocol which is in the standard. In this exercise, some questions are specific to BAC.

BAC makes no use of any pk/sk pair. It works as follows: the reader and the chip derive $K_{\mathsf{init}} = \mathsf{KDF}(w)$ using a key derivation function, select some random nonces N_r (for the reader) and N_c (for the chip) and some keys K_r (for the reader) and K_c (for the chip); the chip sends N_c in clear to the reader; the reader sends (N_r, N_c, K_r) securely (using K_{init}) to the chip; the chip checks that N_c is correct and sends (N_c, N_r, K_c) securely (using K_{init}) to the reader; the reader checks that N_r is correct; the reader and the chip derive $K = \mathsf{KDF}(K_r, K_c)$.

$$\begin{array}{rcl}
\textbf{Reader} & & \textbf{Chip} \\
\text{input: } w & & \text{input: } w \\
 & & \\
K_{\mathsf{init}} \leftarrow \mathsf{KDF}(w) & & K_{\mathsf{init}} \leftarrow \mathsf{KDF}(w) \\
\text{pick } N_r, K_r & & \text{pick } N_c, K_c \\
 & \xleftarrow{\quad N_c \quad} & \\
 & \xrightarrow{\mathsf{Enc}_{K_{\mathsf{init}}}(N_r,N_c,K_r)} & \\
 & & \text{decrypt, check } N_c \\
 & \xleftarrow{\mathsf{Enc}_{K_{\mathsf{init}}}(N_c,N_r,K_c)} & \\
\text{decrypt, check } N_r & & \\
K \leftarrow \mathsf{KDF}(K_r, K_c) & & K \leftarrow \mathsf{KDF}(K_r, K_c) \\
 & & \\
\text{output: } K & & \text{output: } K
\end{array}$$

Q.1 If the password-based key agreement protocol makes no use of any pk/sk pair like in BAC, prove that when the holder shows his biometric passport to someone (for instance, at the hotel check in counter), this person can easily copy the passport. How could this be fixed?

Q.2 If an adversary has obtained w (by whatever means), what is the threat for the holder of the passport? Describe a possible scenario.

Q.3 If we use BAC as a password-based key agreement protocol, prove that the password w and all transmitted data can be recovered in clear with a passive offline exhaustive search. How could we replace BAC to avoid this attack?

Q.4 One difference between a regular identity document (with ID and a picture bio printed) with an official stamp σ and a digital document $(\mathsf{ID}, \mathsf{bio}, \mathsf{pk})$ with a digital signature σ is that we cannot use a photocopy of the stamped document as a proof, whereas we can use an electronic copy of the digital signed document like the original one.

What is the potential threat related to this difference? Explain the related cryptographic notion and a possible scenario. How could this problem be fixed?