Cryptography and Security — Final Exam

Serge Vaudenay

20.1.2015

- 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 Hidden Collisions in DSA

We recall the DSA signature scheme:

Public parameters (p, q, g): pick a 160-bit prime number q, pick a large a random until $p = aq + 1$ is prime, pick h in $\mathbf{Z}_p^*$ and take $g = h^a \bmod p$ until $g \neq 1$.
Set up: pick $x \in \mathbf{Z}_q$ (the secret key) and compute $y = g^x \bmod p$ (the public key).
Signature generation for a message M: pick a random $k \in \mathbf{Z}_q^*$, compute

$$r = (g^k \bmod p) \bmod q \quad s = \frac{H(M) + xr}{k} \bmod q$$

the signature is $\sigma = (r, s)$.
Verification: check that $r = \left(g^{\frac{H(M)}{s} \bmod q} y^{\frac{r}{s} \bmod q} \bmod p\right) \bmod q$.

The hash function H is the SHA-1 standard. The output of H is a binary string which is implicitely converted into an integer. DSA was standardized by NIST with a usual suspicion that the NSA was behind it. It could be the case that some specific choices for (p, q, g) could indeed hide some special property making an attack possible. This is what we investigate in this exercise.

Q.1 What is the complexity of finding m and m' such that $m \neq m'$ and $H(m) = H(m')$?
Q.2 Describe a chosen-message signature-forgery attack based on the fact that an adversary knows two messages m and m' such that $m \neq m'$ and $H(m) = H(m')$.
Q.3 Describe a chosen-message signature-forgery attack based on the fact that an adversary knows two messages m and m' such that $m \neq m'$ and $q = H(m) - H(m')$ (with the integer subtraction).
Propose a way for the NSA to generate public parameters (p, q, g) in such a way that it can later perform a forgery attack for a suitable message.
Q.4 To put more confidence, NIST added a way to certify that (p, q, g) were honestly selected. For this, we shall provide together with the public parameters a value seed such that

$$q = (H(\mathsf{seed}) \oplus H(\mathsf{seed} + 1)) \vee 2^{159} \vee 1$$

where $\oplus$ denotes the bitwise XOR, $\vee$ denotes the bitwise OR, and $+$ is the regular addition of integers. I.e., q is the XOR between $H(\mathsf{seed})$ and $H(\mathsf{seed} + 1)$ after which the least and the most significant bits are forced to 1 so that $2^{159} \leq q < 2^{160}$ and q is odd.

Propose a way to construct (seed, p, q, g) such that an attack is still possible.
HINT: take $m = \mathsf{seed}$ and $m' = \mathsf{seed} + 1$ and estimate the probability that $|H(m) - H(m')| = q$ for seed random, by looking at the propagation of carry bits in the subtraction.
HINT^2: you may skip this question.

2 DSA With Related Randomness

We recall the DSA signature scheme:

Public parameters (p, q, g): pick a 160-bit prime number q, pick a large a random until $p = aq + 1$ is prime, pick h in $\mathbf{Z}_p^*$ and take $g = h^a \bmod p$ until $g \neq 1$.
Set up: pick $x \in \mathbf{Z}_q$ (the secret key) and compute $y = g^x \bmod p$ (the public key).
Signature generation for a message M: pick a random $k \in \mathbf{Z}_q^*$, compute

$$r = (g^k \bmod p) \bmod q \quad s = \frac{H(M) + xr}{k} \bmod q$$

the signature is $\sigma = (r, s)$.
Verification: check that $r = \left(g^{\frac{H(M)}{s} \bmod q} y^{\frac{r}{s} \bmod q} \bmod p\right) \bmod q$.

Sampling the randomness k to sign is critical. This exercise is about bad sampling methods.
In what follows, we consider two messages m_1 and m_2, a signature (r_i, s_i) for message m_i using the randomness k_i, $i = 1, 2$.

Q.1 Sometimes, random sources are not reliable and produce twice the same value. If $k_1 = k_2$, show that from the values of $p, q, g, y, r_1, s_1, r_2, s_2, m_1, m_2$ we can recover x.
Q.2 To avoid the previous problem, a crypto apprentice decides to sample k based on a counter. Redo the previous question with $k_2 = k_1 + 1$.
Q.3 To avoid the previous problem, a crypto apprentice decides to sample k by iterating an affine function. Redo the previous question for $k_2 = \alpha k_1 + \beta$ with α and β known.
Q.4 To avoid the previous problem, a crypto apprentice decides to sample k by iterating an quadratic function. Redo the previous question for $k_2 = \alpha k_1^2 + \beta k_1 + \gamma$ with α, β, and γ known.

3 Reset Password Recovery

We consider a non-uniform distribution D of passwords. Passwords are taken from a set $\{k_1, \ldots, k_n\}$ and each password k_i is selected with probability $\Pr_D[k_i]$. (We omit the subscript D when there is no ambiguity in the distribution.) For simplicity, we assume that $\Pr[k_1] \geq \Pr[k_2] \geq \cdots \geq \Pr[k_n]$. We consider a game in which a cryptographer apprentice plays with a black-box device which has two buttons — a *reset* button and a *test* button — and a keyboard.

- When the player pushes the reset button, the device picks a new password K, following the above distribution, and stores it into its memory. The game cannot start before the player pushes this button.
- The player can enter an input w on the keyboard and push the test button. This makes the device compare K with w. If $K = w$, the device opens, the player wins, and the game stops. Otherwise, the device remains closed and the player continues.

A strategy is an algorithm that the player follows to play the game. Given a strategy, we let C denote the expected number of times the player pushes the test button until he wins. The goal of the player is to design a strategy which uses a minimal C.

In this exercise, we consider several strategies. To compare them, we use a toy distribution T defined by the parameters a, p and

$$\Pr_T[k_1] = \cdots = \Pr_T[k_a] = \frac{p}{a} \quad , \quad \Pr_T[k_{a+1}] = \cdots = \Pr_T[k_n] = \frac{1-p}{n-a}$$

and assuming that $\frac{p}{a} \geq \frac{1-p}{n-a}$.

Q.1 We consider a strategy in which the player always pushes the reset button before pushing the test button. For a general distribution D, give an optimal strategy and the corresponding value of C.
Apply the general result to the toy distribution T.

Q.2 We consider a strategy in which the reset button is never used again after the initial reset. For a general distribution D, give an optimal strategy and the corresponding value of C.
Apply the general result to the toy distribution T.

Q.3 For $n = 3$ and $a = 1$, propose one value for p in the toy distribution T so that the strategy in Q.2 is better and one value for p so that the strategy in Q.1 is better.
We recall that we must have $\frac{p}{a} \geq \frac{1-p}{n-a}$.

Q.4 We consider a strategy in which the player always pushes the reset button after m tests have been made since the last reset. For a general distribution D, give an optimal strategy and the corresponding value of C.
Check that your result is consistent with those from Q.1 and Q.2 with $m = 1$ and $m = n$.

4 A Bad EKE with RSA

In this exercise we want to apply the EKE construction with the RSA cryptosystem and the AES cipher to derive a password-based authenticated key exchange protocol (PAKE). For that, Alice and Bob are assumed to share a (low-entropy) password w. The protocol runs as follows:

$$\begin{array}{rcl}
\textbf{Alice} & & \textbf{Bob} \\
\text{password: } w & & \text{password: } w \\
& & \\
\mathsf{RSAGen} \to (\mathsf{sk}, N),\ B_1\|\cdots\|B_8 \leftarrow N & & \\
C_1\|\cdots\|C_8 \leftarrow \mathsf{AESCBC}_w(B_1\|\cdots\|B_8) & \xrightarrow{C_1\|\ldots\|C_8} & B_1\|\cdots\|B_8 \leftarrow \mathsf{AESCBC}_w^{-1}(C_1\|\cdots\|C_8) \\
& & N \leftarrow B_1\|\cdots\|B_8,\ \text{pick } K \in \{0,1\}^{128} \\
& & \Gamma \leftarrow \mathsf{RSAEnc}_{N,3}(K),\ B'_1\|\cdots\|B'_8 \leftarrow \Gamma \\
B'_1\|\cdots\|B'_8 \leftarrow \mathsf{AESCBC}_w^{-1}(C'_1\|\cdots\|C'_8) & \xleftarrow{C'_1\|\cdots\|C'_8} & C'_1\|\cdots\|C'_8 \leftarrow \mathsf{AESCBC}_w(B'_1\|\cdots\|B'_8) \\
\Gamma \leftarrow B'_1\|\cdots\|B'_8,\ K \leftarrow \mathsf{RSADec}_{\mathsf{sk}}(\Gamma) & & \\
& & \\
\text{output: } K & & \text{output: } K
\end{array}$$

Here are some explanations:

- Alice generates an RSA modulus N such that $\mathsf{gcd}(3, \varphi(N)) = 1$. This modulus is supposed to have exactly 1024 bits. The modulus N is written in binary and splits into 8 blocks $N = B_1\|\cdots\|B_8$. The blocks $B_1, \ldots, B_8$ are then encrypted with AES in CBC mode with IV set to the zero block and the key set to w. The obtained ciphertext blocks $C_1, \ldots, C_8$ are sent to Bob.

- Bob decrypts $C_1, \ldots, C_8$ following the AES-CBC decryption algorithm with IV set to the zero block and the key set to w. He recovers $B_1, \ldots, B_8$ and can reconstruct N. He picks a random 128-bit key K and computes the RSA-OAEP encryption of K with key N and $e = 3$. He then obtains a ciphertext Γ. This is split into 8 blocks $\Gamma = B'_1 \| \cdots \| B'_8$ and the blocks $B'_1, \ldots, B'_8$ are then encrypted with AES in CBC mode with IV set to the zero block and the key set to w. The obtained ciphertext blocks $C'_1, \ldots, C'_8$ are sent to Alice.
- Alice decrypts $C'_1, \ldots, C'_8$ following the AES-CBC decryption algorithm with IV set to the zero block and the key set to w. She recovers $B'_1, \ldots, B'_8$ and can reconstruct Γ. She applies the RSA-OAEP decryption on Γ with her secret key and obtains K.

So, Alice and Bob end the protocol with the secret K.

Q.1 Assume (*only in this question*) that we use plain RSA instead of RSA-OAEP. Show that Eve can easily recover w and K in a *passive* attack with a single execution of the protocol. HINT: show that the plain RSA decryption of Γ is easy in this case.

Q.2 Propose a *passive* attack allowing Eve to deduce the password w after a few executions of the protocol. Estimate the number of executions needed to recover a password with less than 48 bits of entropy with a high probability.
HINT: N is not an arbitrary bitstring. You could think of eliminating some password guesses.