Cryptography and Security — Final Exam

Serge Vaudenay

25.1.2012

- duration: 3h00
- no documents are allowed
- a pocket calculator is allowed
- communication devices are not allowed
- exam invigilators will not answer any technical question during the exam
- answers to every exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

1 Security Issue in ECDSA

In Sony PS3, the bootup code can be changed when it comes from a valid signature from the manufacturer. The signature scheme is ECDSA. We briefly recall the scheme here.

The public key consists of a prime number n, a finite field $\mathsf{GF}(q)$, an elliptic curve over this field, a generator G of order n, and another point Q. The secret key is an integer $d \in \mathbf{Z}_n^*$ such that $Q = dG$. To sign a message M, the signer picks $k \in \mathbf{Z}_n^*$, computes the point $(x_1, y_1) = kG$, then $r = \bar{x}_1 \bmod n$ given a function $x \mapsto \bar{x}$ from $\mathsf{GF}(q)$ to $\mathbf{Z}$, and finally $s = \frac{H(M)+dr}{k} \bmod n$ given a hash function H. If $r = 0$ or $s = 0$, the signer restarts the computation until $r \neq 0$ and $s \neq 0$. The signature is the pair (r, s). To verify a signature (r, s) for a message M, the verifier checks that $Q \neq \mathcal{O}$, that Q lies on the curve, that $nQ = \mathcal{O}$, and that $r \in \mathbf{Z}_n^*$. Then, he computes $u_1 = \frac{H(M)}{s} \bmod n$, $u_2 = \frac{r}{s} \bmod n$, and $(x_1, y_1) = u_1 G + u_2 Q$, and finally checks that $r = \bar{x}_1 \bmod n$.

Q.1 ECDSA manipulates values of different *types* such as *points*, *field elements*, *integers*, etc. What are the types of k, r, s, y_1, $H(M)$? What is $\mathcal{O}$?

Q.2 What kind of finite fields can we use in practice? Cite at least two and briefly explain how to perform computations in these structures.

Q.3 If a key is valid and a signature is produced by the signing algorithm, show that the verification algorithm will accept the signature.

Q.4 Why is it hard to recover the secret key given the public key?

Q.5 For some reasons, the manufacturer produced signatures for different codes using the same random k. Given two codes M and M' and their signatures (r, s) and (r', s'), respectively, show that an adversary can recover d.

2 Hard Disk Encryption

A hard disk is made of sectors of various length (e.g., 4 096 bytes). We want to encrypt data on the disk using the following constraints:

- we want security (no information leakage);

- we want to use symmetric encryption with a single secret key K for the entire hard disk;
- we prefer to use a block cipher;
- we want to be able to access or update a random piece of information without having to process an entire sector; and
- encryption should be "in-place", i.e., ciphertexts must not be larger than plaintexts.

Q.1 Let ℓ be the block length in bits for the block cipher. We assume that each sector has a length L which is a multiple of ℓ. If i is the index of a sector and j is the index of a block in the sector, we let $x_{i,j}$ denote the plaintext block we would have had at position (i, j) with an unencrypted hard disk. Further, we let $y_{i,j}$ denote the ciphertext block we have in the encrypted hard disk.
What is the value of ℓ in the case of AES? Which mode of operation could we propose to meet all the requirements?

Q.2 We still assume that each sector has a length L which is a multiple of ℓ. We define the XTS mode by having a key K composed of two subkeys $K = (K_1, K_2)$ and by having

$$y_{i,j} = \mathsf{Enc}_{K_1}(x_{i,j} \oplus t_{i,j}) \oplus t_{i,j} \quad \text{with} \quad t_{i,j} = \alpha^j \times \mathsf{Enc}_{K_2}(i),$$

where α is a constant and $\alpha^j \times u$ is defined by $\mathsf{GF}(2^\ell)$ operations. Explain how to decrypt and show that it meets all requirements. What is the problem if L is not a multiple of ℓ?

Q.3 We assume that there is at most one incomplete block per sector and that the size of the sector is $L \geq \ell$. We assume that there are n_i blocks in sector i, that $j \in \{1, \ldots, n_i\}$, and that the incomplete block (if any) is the one of index n_i. We use the XTS mode from the previous question with the *ciphertext stealing* technique for the special blocks of index $n_i - 1$ and n_i. Ciphertext stealing consists of using a special rule to compute y_{i,n_i-1} and y_{i,n_i} from x_{i,n_i-1} and x_{i,n_i}:

- if the size of x_{i,n_i} is ℓ, proceed as in the previous question;
- otherwise, split $\mathsf{Enc}_{K_1}(x_{i,n_i-1} \oplus t_{i,n_i-1}) \oplus t_{i,n_i-1}$ into $y_{i,n_i} \| u$, where y_{i,n_i} is an incomplete block, having the same length as x_{i,n_i}, and u is the leftover information in the block. Then, $y_{i,n_i-1} = \mathsf{Enc}_{K_1}((x_{i,n_i} \| u) \oplus t_{i,n_i}) \oplus t_{i,n_i}$. (The $\|$ symbol denotes the concatenation operation.)

Explain how to decrypt and show that it meets all requirements.

3 Attack on 2K-3DES

Q.1 What are the block length and the key length in DES? What is the complexity of key recovery exhaustive search in terms of *data*, *known plaintexts* versus *chosen ciphertexts*, *memory*, and *time*?

Q.2 Double DES is defined by

$$y = \mathsf{DES}_{K_1}\left(\mathsf{DES}_{K_2}(x)\right).$$

Explain how the meet-in-the-middle attack works. What is its complexity in terms of *data*, *known plaintexts* versus *chosen ciphertexts*, *memory*, and *time*?

Q.3 Two-key triple DES is defined by

$$y = \mathsf{DES}_{K_1}\left(\mathsf{DES}_{K_2}^{-1}\left(\mathsf{DES}_{K_1}(x)\right)\right).$$

By preparing a dictionary of all $(\mathsf{DES}_k^{-1}(0), k)$ pairs, show that we can break this using many chosen plaintexts and within a time/memory complexity similar to in the previous question.

Hint: Make an exhaustive search on K_1, i.e., guess K_1, do something, then use the dictionary to recover K_2.

4 Collisions with a Subset

In a classroom, we have x female students and y male students. We assume that their birthday is uniformly distributed in a calendar of N possible dates, e.g., $N = 365$.

Q.1 Let p_{xx} denote the *exact* probability, that there are two different female students with the same birthday. Express p_{xx} in terms of N and x.

Q.2 Let $p_{xy|\neg xx}$ denote the *exact* probability, that there is at least one female-male pair of students who share the same birthday conditioned to that female students have pairwise different birthdays. Express $p_{xy|\neg xx}$ in terms of N, x, and y.

Q.3 Show that $p_{xy|\neg xx} \approx 1 - e^{-\frac{xy}{N}}$.

Q.4 Based on the previous computations, what is the *exact* probability $p_{x\star}$ that at least one female student shares the same birthday with another student (either female or male)?

Q.5 Show that $p_{x\star} \approx 1 - e^{-\frac{x(x+2y)}{2N}}$.
Hint: $p_{xx} \approx 1 - e^{-\frac{x^2}{2N}}$.

Q.6 In a community of n_u users each having a password, we assume that there is a public directory for the hash of the passwords. We consider an attacker who tries to find password matches with the existing database of n_u password hashes. He is allowed to try n_t many random passwords and hash them. We say that he succeeds if he gets any match. That is to say, he succeeds if either he finds at least one password with a hash in the directory, or if he finds two users having the same password hash in the directory. What is his success probability?

5 Secure Communication Across the Röstigraben

Warning: this exercise asks you to propose a real solution for a real problem. You are requested to precisely describe your proposed solution so that we could assess on correctness, feasibility, efficiency, and security. Take this exercise as if it was for a hiring interview for an engineer position.

You want to communicate securely with your friend in Zurich, but you forgot to prepare for it the last time you met. Fortunately, you are making MSc studies with courses in cryptography, so you are familiar with communication systems and computers, and so is your friend.

Q.1 How would you generate a private/public key pair on your computer?

Q.2 How would you and your friend *securely* exchange your public keys?

Q.3 How would you use public keys to set up a symmetric key with your friend?

Q.4 How would you implement a secure communication channel based on this key?

Q.5 Under which assumptions would your system be secure?