

Cryptography and Security — Final Exam

Serge Vaudenay

17.1.2017

- 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 Stealing Bitcoins

We recall the ECDSA signature scheme.

- We are given a point G of an elliptic curve and its prime order n .
- A secret key is a value $d \in \mathbf{Z}_n$ and the public key is the point $Q = dG$.
- To sign a message M , we pick a random $k \in \mathbf{Z}_n^*$ and we compute $r = \overline{(kG)_1} \bmod n$, where $(kG)_1$ denotes the x -coordinate of kG and $\overline{(kG)_1}$ denotes its conversion into an integer; we compute $s = \frac{H(M)+dr}{k} \bmod n$, where $H(M)$ is the digest of M ; the signature of M is (r, s) .
- To verify a signature (r, s) of a message M for Q , we just compare r with a function V of (G, n, Q, r, s, M) .

Q.1 Say what is the verification function V .

Q.2 Assuming that a signing algorithm is implemented using a terrible random number generator (namely, with one for which the generated number often repeats), show that given two signed messages (M, r, s) and (M', r', s') for the public key Q , an adversary may extract the secret key d .

Q.3 We recall that a bitcoin transaction for an account Q is a signature of a will to collect the UTXO $utxo_1, \dots, utxo_t$ owned by Q and split them to some accounts Q_1, \dots, Q_u . Assuming that a user has implemented his ECDSA signing algorithm using a terrible random number generator, show how we can steal his bitcoins.

2 Kleptography

Alice wants to communicate securely with Bob. For that, she buys a highly secure tamperproof device from the company `mole.com` and uses it to communicate. Upon reset, this device generates its own RSA secret key with random primes p and q , modulus n , and exponents e and d . In this implementation, the exponent e is random and of the same size as the modulus. It outputs n and e . Then, when we input a ciphertext, it decrypts it and returns the plain message.

Q.1 Suggest a modulus length for good security and describe the algorithm to generate e .

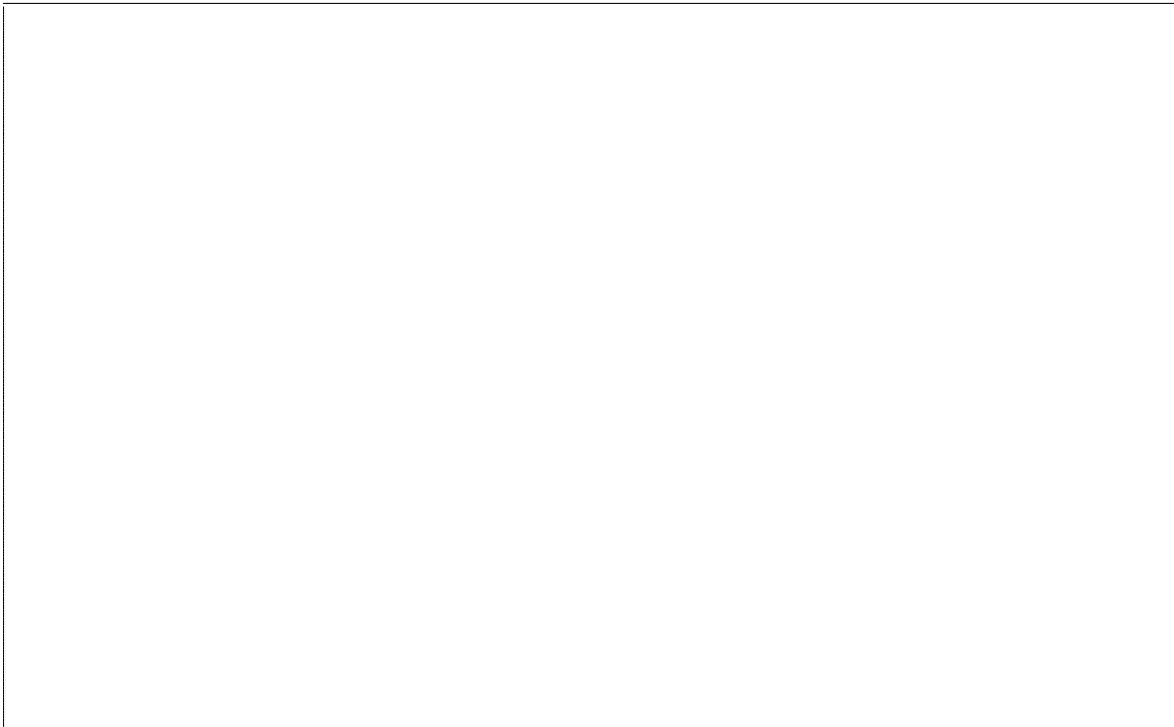
Q.2 The company `mole.com` is hiding a trapdoor to be able to decrypt messages. For this, there is a symmetric secret key t which is put inside the device and the exponent e is chosen of form $\text{SymEnc}_t(p) \parallel \text{random}$.
Explain how `mole.com` can decrypt messages sent to Alice.

Q.3 Some agencies from the “axis of evil” (in the sense of G.W. Bush) succeed to reverse engineer devices from `mole.com`. Show that this creates a major national security issue for the country in which these devices are sold.

Q.4 Using asymmetric encryption, propose a new generation of mole.com devices which allows the company to continue to decrypt messages without risking a security break in the case of reverse engineering.



Q.5 In the above question, when the selected asymmetric encryption is RSA, observe that due to moduli sizes, this decreases the security of the encryption. Propose a way to fix this.



3 AES-GCM Issues

We recall the GCM mode of AES. We modified it a bit for simplicity in this exercise: we assume no associated data, all messages have a length multiple of 128 bits, and the authentication tag has 128 bits. To encrypt a message P with an AES key K and a 96-bit nonce IV , we split it into m 128-bit blocks $P = (P_1, \dots, P_m)$ and run the following algorithm (written in pseudocode).

- 1: $J_i = IV \parallel (i + 1 \bmod 2^{32})_{32}$, $i = 0, \dots, m$, where x_{32} is the binary representation of x in 32 bits
- 2: $C = (C_1, \dots, C_m)$ where $C_i = P_i \oplus \text{AES}_K(J_i)$
- 3: $H = \text{AES}_K(0^{128})$ (called the *authentication key*)
- 4: $S = C_1 H^m \oplus \dots \oplus C_m H$ (with multiplications in $\text{GF}(2^{128})$)
- 5: $T = S \oplus \text{AES}_K(J_0)$
- 6: the output is (C, T)

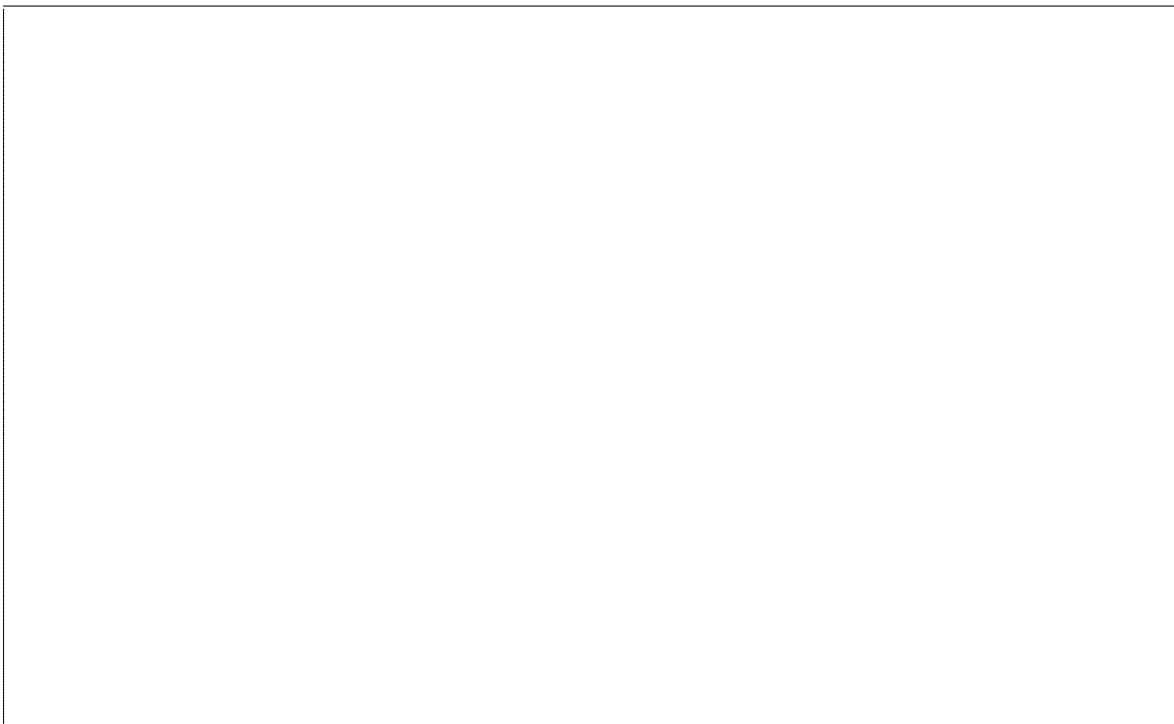
Q.1 Give the description of decryption/authentication in pseudocode.

Q.2 Assuming that a user encrypts a message with m larger than 2^{32} , show how an adversary can recover the XOR of some plaintext blocks from the ciphertext (C, T) .

Q.3 Assuming that a user encrypts two messages P and P' with the same nonce IV , show how an adversary can recover a set of small cardinality which contains the authentication key H . (For simplicity, we assume that P and P' have the same length.)



Q.4 If the adversary knows a set of small cardinality to which H belongs, show how he can decrypt any ciphertext (C, T) with a nonce IV by a chosen ciphertext attack.




4 Prime Reuse in RSA

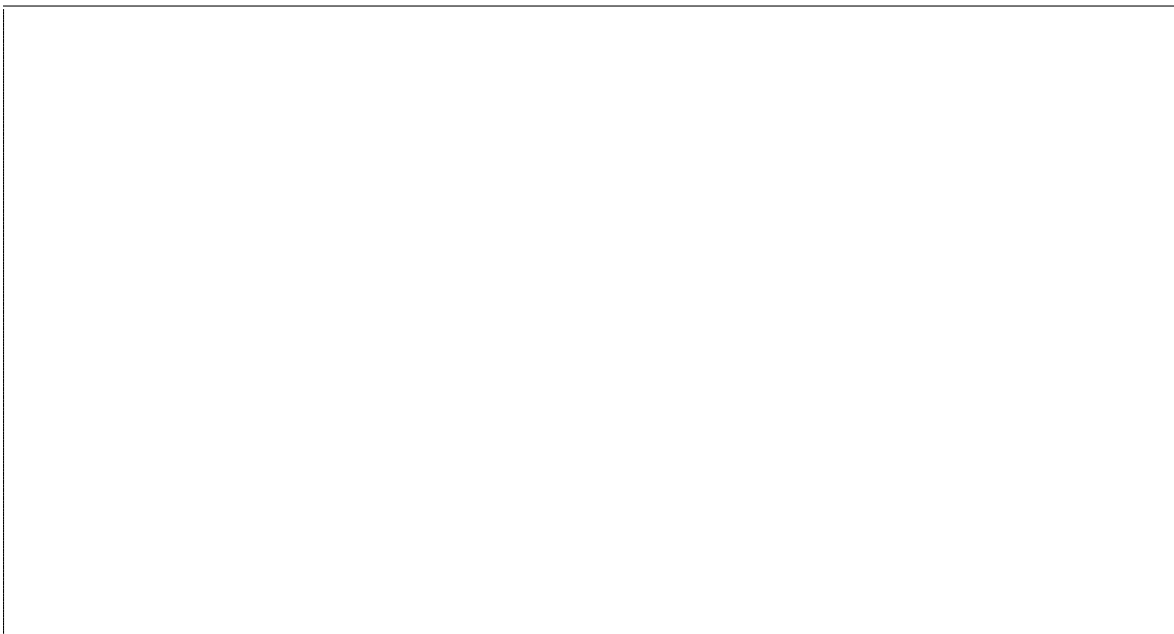
In this exercise, we consider a pool of D RSA keys with a modulus length of s bits.

We recall that the probability of a uniformly distributed random number in $\{1, \dots, n\}$ to be prime is approximately $\frac{1}{\ln n}$.

- Q.1** Say how to check if two different RSA moduli use a prime factor p in common and why it is a security problem.



- Q.2** Using truly random prime numbers, estimate the probability that there exist two RSA keys on the Internet which have a prime factor in common (or estimate the number of pairs with a common prime factor). Justify your answer precisely.



Q.3 By scanning public keys over the Internet, one can find about $D = 11\,170\,883$ keys of size $s = 1\,024$. We observed that 16 717 RSA keys share a common prime factor. What can we deduce?

5 Subgroup Issues in the Diffie-Hellman Protocol

Let g be an element of a (multiplicative) Abelian group G and let $\langle g \rangle$ denote the subgroup of G generated by g . Let q denote the order of g . Let n denote the order of G . We consider the Diffie-Hellman protocol in which Alice picks a secret $x \in \mathbf{Z}_q^*$, computes $X = g^x$, sends X to Bob. Bob picks a secret $y \in \mathbf{Z}_q^*$, computes $Y = g^y$, sends Y to Alice. Alice checks that $Y \in \langle g \rangle$ and computes $K = Y^x$. Bob checks that $X \in \langle g \rangle$ and computes $K = X^y$.

Let B be a given bound. Given $q = q_s q_l$ with $q_s = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ where the p_i are pairwise different “small” primes (i.e. $p_i \leq B$) and q_l has no “small” factor except 1, we denote $\mathcal{F}_B(q) = \{(p_1, \alpha_1), \dots, (p_r, \alpha_r)\}$.

We recall that we have “efficient” (i.e. polynomial in $B + \log q$) partial factoring algorithms to compute \mathcal{F}_B .

We also recall that if p is a “small” prime (i.e. $p \leq B$) and $g' \in G$ is an element of order p^α , then there is an “efficient” (i.e. polynomial in αB) algorithm to compute the discrete logarithm in $\langle g' \rangle$. More precisely, there is an algorithm \mathcal{L}_0 such that $\mathcal{L}_0(B, p, \alpha, g', g'^x) = x \bmod p^\alpha$.

- Q.1** If g has order q and $(p_1, \alpha_1) \in \mathcal{F}_B(q)$, show that there is an “efficient” algorithm to compute $x \bmod p_1^{\alpha_1}$ from g^x . More precisely, show that there is an algorithm \mathcal{L}_1 such that $\mathcal{L}_1(B, q, p_1, \alpha_1, g, g^x) = x \bmod p_1^{\alpha_1}$ for any x . Justify your answer.

- Q.2** If g has order $q = q_s q_l$ with $q_s = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ where $\mathcal{F}_B(q) = \{(p_1, \alpha_1), \dots, (p_r, \alpha_r)\}$, show that there is an “efficient” algorithm to compute the modulo q_s part of the discrete logarithm in $\langle g \rangle$. More precisely, show that there is an algorithm \mathcal{L} such that $\mathcal{L}(B, q, g, g^x) = x \bmod q_s$ for any x . Justify your answer.

Q.3 With the previous notation, show that if instead of picking x in \mathbf{Z}_q^* Alice picks x uniform in $\{1, \dots, q_s - 1\} \cap \mathbf{Z}_q^*$, then a passive adversary can recover x .



Q.4 We now assume that y is a static key (i.e. Bob runs many sessions of the protocol by using the same value y). We consider that after the Diffie-Hellman protocol, Bob sends the encryption of a known message m (e.g. the null message $m = 0$) with the key $\text{KDF}(X^y)$. Encryption is done using a deterministic symmetric encryption. We write $n = pqr$ where p is a “small” prime (i.e. $p \leq B$). Show that if Bob does not verify that $X \in \langle g \rangle$, an active adversary can recover $y \bmod p$ by maliciously selecting X .

