Security Protocols and Application — Final Exam

Philippe Oechslin and Serge Vaudenay

25.6.2014

- duration: 3h00
- no document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will not answer any technical question during the exam
- the answers to each 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 Fully Homomorphic Encryption

Questions Q.1, Q.2, and Q.3 are independent.

Q.1 We consider a fully homomorphic encryption scheme

$$\mathsf{Gen} \to (\mathsf{sk},\mathsf{pk}) \quad , \quad \mathsf{Enc}_{\mathsf{pk}}(x) \to y \quad , \quad \mathsf{Dec}_{\mathsf{sk}}(y) = x \quad , \quad \mathsf{Eval}_{\mathsf{pk}}(f,y) \to z$$

where $x \in \mathbf{Z}_2^n$ and f is a function from $\mathbf{Z}_2^n$ to itself (defined by a circuit with universal gates). We assume that we always have

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}_{\mathsf{pk}}(f,\mathsf{Enc}_{\mathsf{pk}}(x))) = f(x)$$

If $\mathsf{Dec}_{\mathsf{sk}}$ is a one-to-one mapping, show that the encryption is insecure: it is easy to decrypt with pk only.
HINT: use $f_i(x) = (x_i,\ldots,x_i)$.

Q.2 We consider a fully homomorphic encryption scheme

$$\mathsf{Gen} \to (\mathsf{sk},\mathsf{pk}) \quad , \quad \mathsf{Enc}_{\mathsf{pk}}(x) \to y \quad , \quad \mathsf{Dec}_{\mathsf{sk}}(y) = x \quad , \quad \mathsf{Eval}_{\mathsf{pk}}(f,y) \to z$$

We assume that we always have

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}_{\mathsf{pk}}(f,\mathsf{Enc}_{\mathsf{pk}}(x))) = f(x)$$

We consider the following protocol between $A(\mathsf{sk})$ and $B(\mathsf{pk})$, where A holds some secret sk and both participant hold pk, a function f, and some ciphertext $y = \mathsf{Enc}_{\mathsf{pk}}(x)$ for some unknown value x:

1: **for** $i = 1$ to n **do**
2: B flips a coin b_i
3: if $b_i = 0$ then B takes $z_i = \mathsf{Enc}_{\mathsf{pk}}(0)$, otherwise B takes $z_i = \mathsf{Eval}_{\mathsf{pk}}(f,y)$
4: B sends z_i to A
5: A computes $t_i = \mathsf{Dec}_{\mathsf{sk}}(z_i)$
6: if $t_i = 0$, A sets $b'_i = 0$, otherwise, A sets $b'_i = 1$

7: A sends $\mathsf{Commit}(b'_i)$ to B
8: B reveals b_i and how he computed z_i
9: A checks that z_i was well computed
10: A opens his commitment
11: B checks that the commitment is correct and gets b'_i
12: B checks that $b_i = b'_i$
13: **end for**

If any verification fails the protocol aborts.

Q.2a When correctly executed and n large enough, show that the protocol succeeds *if and only if* $f(x) \neq 0$.

Q.2b Show that if b_i is always set to 1, then A can cheat and make the protocol succeed even for $f(x) = 0$.

Q.2c Show that if Step 9 is not done, then B can cheat and learn the value of x.

Q.3 We consider an algorithm $\mathsf{Gen} \to (\mathsf{sk}, \mathsf{pk})$ such that pk defines two commutative groups G and Q, where Q consists of half of the elements of G. (We use multiplicative notations.) We assume that it is easy to compute products, to compare two elements, to find the unit 1, and to sample elements with uniform distribution in Q or in G. The public key pk further defines a fixed element $g \in G$ which is *not* in Q. We further assume that the secret key sk makes it easy to test if an element is in Q or not, but that this operation is hard when knowing only pk.
Given a bit $b \in \mathbf{Z}_2$, we define $\mathsf{Enc}_{\mathsf{pk}}(b) = g^b r$ where r is a random element in Q (which is freshly picked to encrypt b) and g is defined by pk. We define $\mathsf{Dec}_{\mathsf{sk}}(c) = 0$ if $c \in Q$ and $\mathsf{Dec}_{\mathsf{sk}}(c) = 1$ otherwise.

Q.3a Given a linear function L from $\mathbf{Z}_2^n$ to $\mathbf{Z}_2$, define an algorithm Eval_L such that

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}_L(\mathsf{Enc}_{\mathsf{pk}}(b_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(b_n))) = L(b_1, \ldots, b_n) \tag{1}$$

for all $b_1, \ldots, b_n \in \mathbf{Z}_2$.
HINT: for all $x \in G$, we have $x^2 \in Q$.

Q.3b Formally prove that Eval_L satisfies (1).

2 Zerocoin, the really anonymous cryptocurrency

Bitcoin

Q.1 Bitcoin uses a proof-of-work function which is used to sign blocks of transactions.
List three properties that the proof-of-work function must have and give a motivation for each property.

Q.2 Recently a group of miners (a mining pool) reached 51% of the total mining power of the network.
- Describe an attack that they could have carried out in order enrich themselves.
- Explain why it is probably not in their interest to carry out such an attack.

Q.3 If a seller does not wait for a confirmation of a transaction to appear in a new block, it may be possible to invalidate a transaction even with very limited or no mining power at all.
Can you describe an attack were a transaction signed by the buyer and propagated by the seller does not end up in the next chain block ?

Q.4 A simple Bitcoin laundering service could be the following: Customers who wish to wash their coins transfer them to the single account of a laundry service. After a random delay the laundry service transfers the coins to a destination account chosen by the customer.
Describe two attacks that a dishonest laundry service could run against its customers.

Zerocoin

Q.5 Zerocoin uses cryptographic accumulators which can store a set of elements. Given only the accumulator, it is not possible to find the elements that it holds. However, given the accumulator and one element, it is possible to prove that the element is in the accumulator.
A simple accumulator can be created with multiplication. You add a value x to the accumulator A by multiplying the value and the accumulator:

$$A_{n+1} = A_n \cdot x$$

- How can you prove that a value v is stored in the accumulator A ?
- What conditions must hold to make it hard to find the elements that are in the accumulator ?
- What makes this accumulator unpractical to implement ?

Q.6 The accumulator suggested for Zerocoin is of the form

$$A_{n+1} = A_n^x \mod P$$

- How must the values for x be chosen for the accumulator to work properly ?
- How can you create a proof that a certain value v is in the accumulator ?

Q.7 When spending a Zerocoin, you have to create a zero knowledge proof (ZKP) that shows that you have the right to spend a coin.
- What is the information that you need to keep when you mint a Zerocoin such that you can generate the ZKP when you want to spend a coin ?
- What exactly does the ZKP prove ?

Q.8 More precisely, Zerocoin uses *non-interactive zero-knowledge proofs of knowledge* (NIZKPoK) and *signatures of knowledge* of messages (ZKSoK).
- Explain the goal of a ZKSoK.
- What is it used for in Zerocoin ?

Q.9 Although Zerocoins are mathematically untraceable there can still arise scenarios where it will be possible to link a spend operation to the mint operation of the same user.
Describe two such scenarios