Advanced Cryptography — Final Exam

Solution

Serge Vaudenay

3.7.2024

- duration: 2h
- any document allowed
- 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

The exam grade follows a linear scale in which each question has the same weight.

1 Soundness of DLEQ NIZK in ROM

This exercise studies the Discrete Logarithm EQuality (DLEQ) proof protocol and the batch version.

Q.1 What are the acronyms "NIZK" and "ROM". Explain what they mean.

> *NIZK is a Non-Interactive Zero-Knowledge protocol. NI: The prover sends a single message to the verifier so that the verifier can check the proof. ZK: The verifier learns no more information than the purpose of the proof (if what is proven is true, the verifier can generate what he learns without interacting).*
> *ROM is the Random Oracle Model. It is a convenient idealized model which enables to prove security when we replace hash functions by truly random ones, implemented by an oracle which we call the "random oracle".*
> *To get all points, it was necessary to say what the acronyms were for, to explain NI and ZK, and to say that RO was to model hash functions.*

Q.2 We assume that a setup phase defined a public group of prime order q. By using the generalized Schnorr Σ-protocol, design a Σ-protocol for the relation R between a tuple of group elements (S,T,U,V) and a witness $w \in \mathbf{Z}_q$ which is true if and only if $U = w \cdot S$ and $V = w \cdot T$.

$$R((S,T,U,V),w) \Longleftrightarrow U = w \cdot S \ \wedge \ V = w \cdot T$$

We later on call it the DLEQ protocol.

We use the generalized Schnorr protocol for the homomorphism $w \mapsto (w \cdot S, w \cdot T)$. The challenge set is $\mathbf{Z}_q$. The prover picks $k \in \mathbf{Z}_q$ uniformly at random, computed $P = k \cdot S$ and $Q = k \cdot T$, and sends the message (P, Q) to the verifier. Upon receiving the challenge e, the prover computes $s = k + ew \bmod q$ and responds with s. The verifier checks that $P + e \cdot U = s \cdot S$ and $Q + e \cdot V = s \cdot T$.
The extractor has input $(S, T, U, V, P, Q, e_0, s_0, e_1, s_1)$ with $e_0 \neq e_1$, s_b being the response to challenge e_b for $b = 0, 1$. It consists of taking $w = \frac{s_1 - s_0}{e_1 - e_0} \bmod q$.
The simulator has input (S, T, U, V, e). It consists of picking $s \in \mathbf{Z}_q$ at random and giving $P = s \cdot S - e \cdot U$ and $Q = s \cdot T - e \cdot V$.
Points were removed if the extractor and the simulator were missing.
Many wrong answers were given for the protocol by giving an AND construction of two Schnorr proofs. However, it proves

$$(\exists w \; U = wS) \wedge (\exists w \; V = wT)$$

instead of $\exists w \; (U = wS \wedge V = wT)$.

Q.3 The Fiat-Shamir transform sets the challenge to $e = H(S, T, U, V, \mathsf{message})$, where message is the first message by the prover, and produces a final "proof" π. If instead we use $e = H(S, T, \mathsf{message})$, show that we can make an algorithm $\mathcal{A}^H(S, T, \mathsf{message}) \to (U, V, \pi)$ making a valid proof for (S, T, U, V), i.e. passing the verification procedure of the Fiat-Shamir transform, even though no witness w may exist.

We let $\mathsf{message} = (P, Q)$. The algorithm sets $e = H(P, Q)$, picks $s \in \mathbf{Z}_q$ at random, computes $U = \frac{1}{e}(s \cdot S - P)$ and $V = \frac{1}{e}(s \cdot T - Q)$, and yields (U, V, π) where $\pi = (P, Q, s)$.
What is interesting to observe is that it is rare that there exists a w such that $U = w \cdot S$ and $V = w \cdot T$ with the obtained values. Hence, the proof π is deceiving the verifier.
The following solution was proposed: since we can set U and V arbitrarily, just pick w and set $U = wS$ and $V = sT$ then follow the prover protocol. However, P and Q are imposed to the adversary in this question so they cannot follow the protocol with them.

Q.4 We want to prove the hardness of forging a valid π. Why shall we better avoid using extractors in such a proof?

In general, using an extractor requires to get the response to two different challenges. In interactive protocols, this is done by "rewinding" the prover, but the arguments become delicate if the adversary interacts with oracles during the rewinding time, as there is no guaranty that the prover will then issue a proof for the same instance and some exponential factors may appear. In non-interactive protocols, the problem is the same when we use the random oracle model, because the protocol now becomes interactive with this oracle. Actually, the forking Lemma is rewinding the prover to some distinguished oracle query. Hence, it is better to apply arguments which do not require rewinding.

There are extraction techniques which do not require rewinding, but they need the protocol to be changed and introduce some complexity overhead. Since our problem is to prove the hardness to forge π, we do not need witness extraction per se. It would be better to rely on a "proof of membership" rather than a "proof of knowledge": only prove that a witness exists rather than a witness is known.

Q.5 We now take the correct Fiat-Shamir transform. We consider an adversary $\mathcal{A}^H$ who interacts with H and is only bounded by a number B of queries but not bounded in terms of computational complexity. The goal of the adversary is to output a tuple (S, T, U, V, π). If the verification passes but there exists no witness w for (S, T, U, V), we say that the adversary wins.

Q.5a Let $\pi = (\mathsf{message}, \mathsf{response})$. Prove that if the final output (S, T, U, V, π) of $\mathcal{A}$ is such that $(S, T, U, V, \mathsf{message})$ was never queried to H, then the probability to win is bounded by $\frac{1}{q}$.

Let $\pi = (P, Q, s)$. We first consider the case where (S, T, U, V, P, Q) was not queried to H by $\mathcal{A}$. Hence, $e = H(S, T, U, V, P, Q)$ is undetermined and the query will be made for the verification. We note that if $U = V = 0$, then $\mathcal{A}$ does not win as the witness $w = 0$ exists. We assume that $U \neq 0$ (the same argument would apply for $V \neq 0$). The number of e values such that $P + e \cdot U = s \cdot S$ is exactly one. The probability that H returns that number is $\frac{1}{q}$. Hence, in this case, the probability to win is bounded by $\frac{1}{q}$.

To get all points, it was necessary to mention the $U = V = 0$ particular case and to explain in other cases why the number of e was limited to one.

Q.5b For any fresh query $(S, T, U, V, \mathsf{message})$ to H, prove that the probability that there exists $\mathsf{response}$ such that the output $(S, T, U, V, \mathsf{message}, \mathsf{response})$ would result in winning is bounded by $\frac{1}{q}$.

For any query (S, T, U, V, P, Q) to H, we let E be the set of all pairs of group elements which can be written in the form $(w \cdot S, w \cdot T)$ for some w. There exists an s making the output (S, T, U, V, P, Q, s) win for $\mathcal{A}$ if and only if $(U, V) \notin E$ and $(P, Q) + e \cdot (U, V) \in E$. The number of e such that this is true is at most 1. Hence, the probability that s exists is bounded by $\frac{1}{q}$.

Q.5c Deduce that for any $\mathcal{A}^H$ limited to B queries, the probability to win is bounded by $\frac{1+B}{q}$.

To win, either one fresh query must lead to winning case, or a no-query must win. There are up to $1+B$ cases. Each case has a winning probability bounded by $\frac{1}{q}$. Hence, the algorithm success probability is bounded by $\frac{1+B}{q}$.
Clarity of the argument was necessary here to get all points.

2 A Simple PRF

We let $D = \{0,1\}^n$ be the domain of the n-bit strings. Given a hash function H from D to itself, we define the function $f_k(x) = H(x \oplus k)$, for $x, k \in D$. We call k a key and x an input to f. We want to show that f is a PRF in the random oracle model. We consider a PRF game in the random oracle model, where the adversary can query H, as well as the oracle which evaluates the function f_k. Let $\mathcal{A}$ be a PRF adversary and let Γ^b be the PRF game with input bit b. In what follows, we prove that $|\Pr[\Gamma^1_{\mathcal{A}} \to 1] - \Pr[\Gamma^0_{\mathcal{A}} \to 1]|$ is negligible.

Q.1 Why shall we indeed consider adversaries making queries to H?

The random oracle model is a trick to replace a real-world hash function H by a random function. In the real world, anybody can evaluate the hash function, because it is specified in the system. Hence, the adversary can as well. If we now outsource the H computation to a random oracle, it does not make sense to restrict the adversary to use it.
Answers such that "because in ROM we give access to the random oracle" received no point.

Q.2 Prove that there exists an adversary $\mathcal{B}$ who never repeats any query to H nor any query to the f-evaluation oracle and such that $\Pr[\Gamma^b_{\mathcal{A}} \to 1] = \Pr[\Gamma^b_{\mathcal{B}} \to 1]$ for every b.

The algorithm $\mathcal{B}$ just simulates $\mathcal{A}$ but intercepts every query. The algorithm $\mathcal{B}$ maintains the list of queries which have been made and the obtained responses. If a query is new, then $\mathcal{B}$ proceeds, makes that query, and forwards the answer to the $\mathcal{A}$ simulation. If a query repeats, then $\mathcal{B}$ checks how it was answered before and gives the same answer to the $\mathcal{A}$ simulation. Clearly, oracles are deterministic in this game so $\mathcal{A}$ sees exactly the same thing as if it was directly interacting with the oracles. Hence, the games give the same outcome.
Answers such as "repeating queries make no sense" or "repeating queries do not bring new information" did not get all the points.

Q.3 Let i be an integer. We define the event E_i that the first i queries made by $\mathcal{B}$ lead to no repetitions on the side of H. Prove that $\Pr[\neg E_{i+1}|E_i] \leq i2^{-n}$.

We know that $\mathcal{A}$ never repeats a query. So the only repeating queries to H must come from two origins: $\mathcal{A}$ and the f-evaluation oracle. If the ith query y to H repeats, it must be that $\mathcal{A}$ queries y to H and $x = y \oplus k$ to the f-evaluation oracle. Hence, k is the XOR of the ith query with one of the previous queries. These are i possible values for k.
When E_i occurs, the first i responses by H can be pre-determined. Hence, the responses are independent from k. The probability that k was selected among the i possible values is bounded by $i2^{-n}$. Hence, $\Pr[\neg E_{i+1}|E_i] \leq i2^{-n}$.

Q.4 We modify the game Γ^b by making H always answer something random and freshly sampled. We denote by $\bar{\Gamma}^b$ the new game. Deduce from the previous question that $|\Pr[\bar{\Gamma}^1_{\mathcal{B}} \to 1] - \Pr[\Gamma^1_{\mathcal{B}} \to 1]| \leq \frac{m^2}{2} 2^{-n}$ and $\Pr[\bar{\Gamma}^0_{\mathcal{B}} \to 1] = \Pr[\Gamma^0_{\mathcal{B}} \to 1]$, where m is the total number of oracle calls.

Let E be the event that there is no repeating query to H. Whenever E holds, Γ^b and $\bar{\Gamma}^b$ are identical, as H would freshly select a new answer at random upon a new query anyway. Furthermore, in the $b = 0$ case with a random function, the f-evaluation oracle makes no query to H so E always occurs, thanks to the construction of $\mathcal{B}$. Hence, $\Pr[\bar{\Gamma}^0_{\mathcal{B}} \to 1] = \Pr[\Gamma^0_{\mathcal{B}} \to 1]$.
In the $b = 1$ case, $\neg E$ occurs if only f-evaluation oracle makes a query to H which collides with one H oracle by $\mathcal{B}$. By the difference Lemma, we have $|\Pr[\bar{\Gamma}^1_{\mathcal{B}} \to 1] - \Pr[\Gamma^1_{\mathcal{B}} \to 1]| \leq \Pr[\neg E]$. Clearly,

$$\neg E \subseteq \bigvee_{i=0}^{m-1} ((\neg E_{i+1}) \wedge E_i)$$

Hence,

$$\Pr[\neg E] \leq \sum_{i=0}^{m-1} \Pr[\neg E_{i+1} \wedge E_i] \leq \sum_{i=0}^{m-1} \Pr[\neg E_{i+1} | E_i] \leq \sum_{i=0}^{m-1} i 2^{-n} \leq \frac{m^2}{2} 2^{-n}$$

To get all points, we had to refer to the difference lemma and not to forget the $b = 0$ case.

Q.5 Prove $\Pr[\bar{\Gamma}^1_{\mathcal{B}} \to 1] = \Pr[\bar{\Gamma}^0_{\mathcal{B}} \to 1]$ and conclude.

The input query to H is not useful anymore in $\bar{\Gamma}^1$, to the f-evaluation oracle in $\bar{\Gamma}^1$ does no longer need to compute it. Clearly, this oracle now does the same in $\bar{\Gamma}^1$ and $\bar{\Gamma}^0$. The game does not use b any more. Hence, $\Pr[\bar{\Gamma}^1_{\mathcal{B}} \to 1] = \Pr[\bar{\Gamma}^0_{\mathcal{B}} \to 1]$.
Given all results, we deduce $|\Pr[\Gamma^1_{\mathcal{A}} \to 1] - \Pr[\Gamma^0_{\mathcal{A}} \to 1]| \leq \frac{m^2}{2} 2^{-n}$.

Q.6 Show that the security bound we obtained is pretty tight by constructing an adversary which (nearly) matches the bound.

We can construct a birthday-bound adversary which uses $m \sim 2^{\frac{n}{2}}$, alternating non-repeating queries to H and the f-evaluation oracles. From the two types of oracles, we obtain two list of queries to H which are likely to collide. Whenever the adversary sees a collision $H(x) = f_k(y)$, the adversary can try to check if $k = x \oplus y$ is coherent with all f-evaluation queries. If k is found, the adversary outputs 1. Otherwise, the answer is 0.
The adversary will succeed in Γ^1 with constant probability. In Γ^0, it is nearly impossible to output 1. Hence, we obtain a constant advantage, which shows that the bound is tight.