Introduction to cryptology
TD#5

2019-W12, ...

Exercise 1: Secure groups for DH

For each of the following groups, state if it can be used to safely implement a Diffie-Hellman key exchange.

— $\mathbb{F}_{2^{3072}}^{\times}$

— $\mathbb{F}_{2^{130}-5}^{\times}$ (note that $2^{130}-5$ is a prime number)

— $\mathbb{F}_{2^{393}17^{91}+1}^{\times}$ (note that $2^{393}17^{91}+1$ is a prime number)

— $\mathbb{F}_{2p+1}^{\times}$, where $2p+1$ and p are both prime (i.e. p is a *Sophie Germain prime*) and $\log(p) \approx 3000$

Exercise 2: Interactive proof of identity

Let $\mathbb{G}$ be a finite group of order N, where the discrete logarithm problem is hard, and g be a generator of a subgroup of $\mathbb{G}$ of prime order p. A *prover* wants to prove to a verifier that s/he knows a number x s.t. $X = g^x$, with $X \in \mathbb{G}$. S/he suggests the following protocol for a *verifier* to check this assertion:

1. The prover picks $r \xleftarrow{\$} [\![0, p-1]\!]$ and sends $R = g^r$ to the verifier
2. The verifier picks a *challenge* $c \xleftarrow{\$} [\![0, p-1]\!]$ and sends it to the prover
3. The prover computes $a = r + cx \mod p$ and sends it to the verifier
4. The verifier computes g^a and accepts the proof if it is equal to RX^c

Q. 1: Show that if the prover indeed knows x, the verifier always accepts the proof.

Q. 2: Why is it important for an honest prover to pick a random r? What would happen if r was easy to predict (say with probability larger than 2^{-40})?

Q. 3: When running the protocol twice, why is it important for the two random numbers r and r' to be distinct?

Q. 4: Show that by picking R and c him/herself, a challenger is able to create a fake run of the protocol that is indistiguishable from a real one. (Hint: try to first pick c and a and compute an R that makes the proof valid.)

Remark: This last property of the above protocol has interesting consequences: it ensures that the prover does not reveal any information about the secret x. The same secret may then be used in many proofs without decreasing the security.

Q. 5: Despite the previous remark, why is there still a limit on the number of times a single secret may be used?

Exercise 3: Random Self-Reducibility of the DLP

In this short exercise, we will see that in prime-order groups, the ability to solve the discrete logarithm problem *on average* allows to solve the problem on any instance. This shows that the worst-case complexity of the problem is not more than the one of average cases (where an average case is defined to be a random problem instance).

Let $\mathbb{G} = \langle g \rangle$ be a finite group of prime order p.

Q. 1: Show how one can construct such a group $\mathbb{G}$ from a $\mathbb{F}_{2p+1}^{\times}$ where p is a Sophie Germain prime.

Q. 2: Let $h = g^a$ be an element whose discrete logarithm we wish to compute. Show that if one knows r, this is equivalent to computing the discrete logarithm of g^{ar}.

Q. 3: Explain why if $r \xleftarrow{\$} [\![0, p-1]\!]$, then $\Pr[g^{ar} = X] = 1/p$ for any $X \in \mathbb{G}$. Why do we need $\mathbb{G}$ to be of prime order for this to be true? (Hint: think of what would happen if $\mathrm{ord}(\mathbb{G})$ were equal to qN' and if one had $a = qA$.)

Q. 4: Assuming you know an efficient algorithm to compute the discrete logarithm of a fraction of 2^{-10} of the elements of $\mathbb{G}$, give an efficient *randomized* algorithm that computes the discrete logarithm of any element of $\mathbb{G}$.

Exercise 4: Three-party Diffie-Hellman using cryptographic pairings

We define a *pairing* $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ over two finite groups $\mathbb{G}_1 = \langle P, Q \rangle$ (noted additively) and $\mathbb{G}_2 = \langle \mu \rangle$ (noted multiplicatively) as being a bilinear, alternating, non-degenerate map. Concretely, this means that $e(S, T + Q) = e(S, T)\, e(S, Q)$ and $e(S + Q, T) = e(S, T)\, e(Q, T)$; $e(T, T) = 1$ and $e(T, S) = e(S, T)^{-1}$; and if $e(S, T) = 1$ for all $S \in \mathbb{G}_1$, then $T = 0$. Furthermore, we say that two elements S and T of $\mathbb{G}_1$ are linearly independent iff. $e(S, T) \neq 1$.

A *cryptographic pairing* is a pairing such that the discrete logarithm problem is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

Q. 1: Show that if P and Q are not linearly independent, then $e(aP, Q) = 1$ for any $a \in \mathbb{N}$, where aP means $\sum_{i=1}^{a} P$.

Q. 2: Show that if P and Q are linearly independent, then $e(aP, bQ) = e(P, Q)^{ab}$, and that this latter value is not constant (in function of a and b).

Q. 3: Let A, B, and C be three actors that wish to share a common secret. One suggests to do the following:

1. Before running the protocol, A, B and C agree on a pairing e and two linearly independent elements P and Q of $\mathbb{G}_1$.
2. Each participant respectively picks a random integer a, b and c (in an appropriate interval) and broadcasts the elements aP and aQ (resp. bP and bQ; cP and cQ) to the others.
3. They all use the pairing e to compute a shared secret.

Show that A, is able to compute the value $e(P, Q)^{abc}$ thanks to the knowledge of a, bP, bQ, cP, cQ, and the same for the two other actors up to an appropriate substitution of the variables.

Explain roughly why this is a secure protocol, assuming that e is a cryptographic pairing.

Note: A typical instantiation of cryptographic pairings is to take $\mathbb{G}_1$ to be a subgroup of the group of points of an elliptic curve and $\mathbb{G}_2$ to be a subgroup of the multiplicative group of a finite field.