ETH zürich

Department of Computer Science

Information Security Lab HS21 – Summary

Author: Yannis Huber

Professors: Prof. Kenny Paterson, Prof. Dennis Hofheinz, Prof. Adrian Perrig, Prof. Srdjan Capkun, Prof. Shweta Shinde

1 Introduction

This document summarises the most important notions acquired during the master's degree inter-focus course Information Security Lab. The course consists of 6 modules each focusing on a different topic in information security, including cryptography, protocol design, network security and system security.

The goal of this course is to provide a broad, hands-on introduction to Information Security, introducing adversarial thinking and security by design as key approaches to building secure systems.

2 Elliptic Curve Cryptography and Lattice Attacks

The first module module introduces elliptic curve cryptography and how it relates to public key cryptography over finite fields. Furthermore, this module also introduces lattice cryptanalysis and how it can be used to attack the ECDSA signature scheme.

2.1 Diffie-Hellman Key Exchange

First let us recall the Diffie-Hellman key exchange protocol in its original implementation. Let p be a prime integer. We consider the multiplicative group of integers modulo p, denoted $\mathbb{Z}_p^*$, of order $p-1$. The two communicating parties, Alice and Bob, are given the prime p and also g which is a carefully chosen[1] generator of the group $\mathbb{Z}_p^*$. The generator g is of order q.

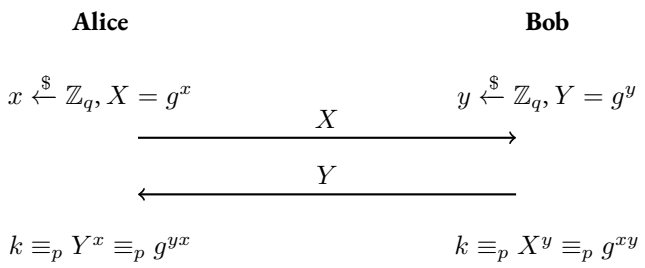


Figure 1: Overview of the Diffie-Hellman key exchange protocol between Alice and Bob.

It is straightforward to see that at the end of the protocol run, Alice and Bob each share the same key k which has never been sent over the insecure channel. Now the fact that this key exchange is indeed secure depends on the discrete logarithm problem.

The Discrete Logarithm Problem (DLP) in $\mathbb{Z}_p$:
Let (p, g, q) be group parameters as defined earlier in this chapter. Set $b \equiv_p g^a$, where a is chosen uniformly at random in $\mathbb{Z}_q$. The goal of the DLP is, given (p, g, q) and b, to find a.

We will see in section 4 that this problem is considered computationally hard and also how it can be proven formally. For now let us assume that solving DLP is hard and that therefore any cryptographic scheme based on DLP can be considered secure. Before we look at how the Diffie-Hellman key exchange can be generalised to other mathematical structures such as elliptic-curves let us quickly remind what elliptic curves are.

2.2 Elliptic Curves

Elliptic curves are defined on a field F. They are represented by a set of pairs $(x, y) \in F \times F$ called points, defined by some equation in x and y. In cryptography, F is chosen as the field of the integers modulo p, for some large prime p (typically 256 bits).

The common form for the equation of an elliptic curve is the following:

$$E = \left\{(x, y) \in F \times F \mid y^2 = x^3 + ax + b\right\} \cup \{\mathcal{O}\}$$

[1] The order of the generator must be big enough. It is good practice to stick with the commonly used values.

Under the condition $4a^3 + 27b^2 \neq 0$ which ensures non singularity of the curve. Notice the special symbol $\mathcal{O}$ which is called the point at infinity that does not have a representation $(x, y) \in F \times F$. This way of representing an elliptic curve is called *short Weierstrass form using affine coordinates*[2].

We can now define an operation on any pair of points on an elliptic curve the obtain a third point, which we will call addition. A geometric interpretation of the addition operation is shown on figure 2.

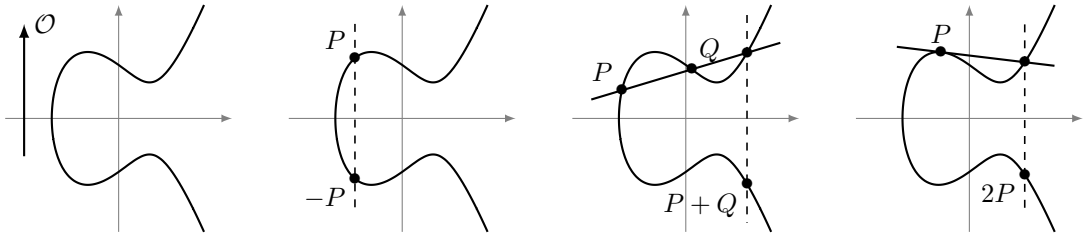
There are 2 cases to distinguish for the addition of two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. First, if $P \neq Q$ then the resulting point $P + Q = (x_3, y_3)$ is calculated as follows[3]:

$$\begin{aligned} \lambda &= \frac{y_2 - y_1}{x_2 - x_1} \\ x_3 &= \lambda^2 - (x_1 + x_2) \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned}$$

If however, $P = Q = (x, y)$, the resulting point $P + P = (x', y')$ is computed as follows:

$$\begin{aligned} \lambda &= \frac{3x^2 + a}{2y} \\ x' &= \lambda^2 - 2x \\ y' &= \lambda(x - x') - y \end{aligned}$$

Each point $P = (x, y)$ also has an inverse denoted $-P = (x, -y)$. The point at infinity $\mathcal{O}$ is its own inverse. Furthermore, the point at infinity acts as the identity element. At this point, it becomes pretty clear that the set of points on an elliptic curve form a group in respect to addition (this requires a proof and is actually pretty simple to conclude from the group axioms).



Neutral element $\mathcal{O}$ | Inverse element $-P$ | Addition $P + Q$ | Doubling $P + P$

Figure 2: Group operations on elliptic curves

3 The TLS 1.3 Protocol

4 Cryptographic Reductions

5 Trusted Execution Environments

6 Software Security

7 DDoS Attacks and Defenses

[2] In practice one has to be careful when implementing elliptic curves under the short Weierstrass form since it can leak sensitive information over various side-channels. There exists alternative forms, for example, the Montgomery form makes it easy to do constant-time scalar multiplication, while Edwards form makes it easier to avoid branching in ECC code.

[3] The equations shown here require the computation of modular inverses which are computationally expensive. We can circumvent this problem by using projective coordinates.