Implementation of elliptic curves algorithms

Roger Truchero Visa
Gerard Donaire Alòs
Joel Solaní Núñez

January 14, 2019

Índex

Definition of elliptic curve

An elliptic curve E over a field K is given by a **Weierstrass** equation,

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, a_i \in \Bbbk \qquad (1)$$

which satisfies, since all of its points are non-singular, the condition that the discriminant Δ is non zero

If the characteristic of $\mathbb{k}$ is different from 2 and 3, this equation can be expressed as

$$y^2 = x^3 + ax + b, a, b \in \mathbb{k} \tag{2}$$

which is called **reduced Weierstrass form**, with discriminant

$$\Delta = -16(4a^3 + 27b^2) \neq 0 \tag{3}$$

Given an elliptic curve E over $\mathbb{k}$, we will denote by E($\mathbb{k}$) the set of points of $\mathbb{k}^2$ which satisfy the curve equation along with the **point at infinity** O, that is,

$$E(\mathbb{k}) = \{(x, y) \in \mathbb{k}^2 \mid y^2 = x^3 + ax + b\} \cup O \qquad (4)$$

Point addition

An operation + can be defined over E(K) by means of the chord–tangent method:

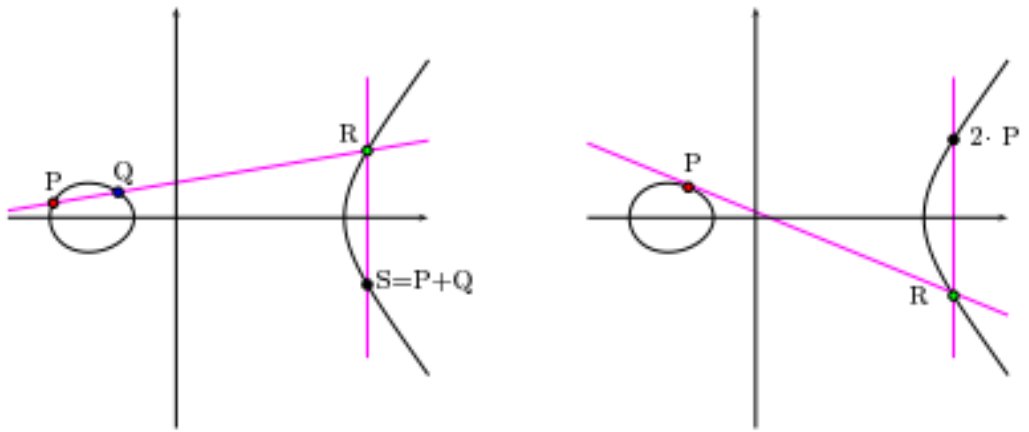

Figure: Adding and doubling points on an elliptic curve

- (E(K), +) is an **abelian group** with O as the zero element. The opposite or symmetric of a point **P = (x, y)** on E(K) is the point **-P = (x, -y)**.

Point multiplication

Let E be an elliptic curve over K.
If P is a point on E(K) and k is an integer, then the point k · P can be defined as:

$$\mathbf{k} \cdot \mathbf{P} = \begin{cases} \text{P} +\ldots^{k}+ \text{P}, & \text{if k} > 0 \\ \text{O}, & \text{if k} = 0 \\ \text{(-P)} +\ldots^{k}+ \text{(-P)}, & \text{if k} < 0 \end{cases} \tag{5}$$

Introduction to elliptic curves

Why elliptic curves?

Since the beginning of public key cryptography there are two major cryptosystems (RSA and El-Gamal) that seem to defeat all attacks. For this reason, these two cryptosystems are the most respected and wildly used public key cryptosystems nowadays.

One can use both cryptosystems for encryption/decryption and digital signatures. All important security standards cover those cryptosystems, so it should be safe to use implementations of them.

The question is now, why do we look for new cryptosystems?

- Is it only mathematical curiosity?
- Do we want to make new products and earn more money?
- Do we know more about ECC and can make cryptosystems with trapdoors to spy out others?
- Are we paid by Certicom?
- or is there a real need for new cryptosystems?

Índex

Public-key encryption

Explanation

Public-key encryption schemes can be used to provide *condifentiality*. Since they are considerably slower than their symmetric-key counterparts, they are typically used only to encrypt small data items such as credit card anumbers and PINs.

Public-key encryption

Steps

A *public-key encryption scheme consists in four algorithms:*

1. A *domain parameter generation algorithm* that generates a set D of domain parameters.
2. A *key generation algorithm* that takes as input a set D of domain parameters and generates key pairs (Q, d).
3. An *encryption algorithm* that takes as input a set of domain parameters D, a public key Q, a plaintext message m, and produces a ciphertext c.
4. A *decryption algorithm* that takes as input the domain parameters D, a private key d, a ciphertext c, and either rejects c as invalid or produces a plaintext m.

Elliptic Curve Integrated Encryption Scheme

Explanation

The **E**lliptic **C**urve **I**ntegrated **E**ncryption **S**cheme (**ECIES**) was proposed by *Bellare* and *Rogaway*, and is a variant of the *ElGamal public-key encryption scheme*. It has been standardized in *ANSI X9.63* and *ISO/IEC 15946-3*, and is in the *IEEE P1363a* draft standard.

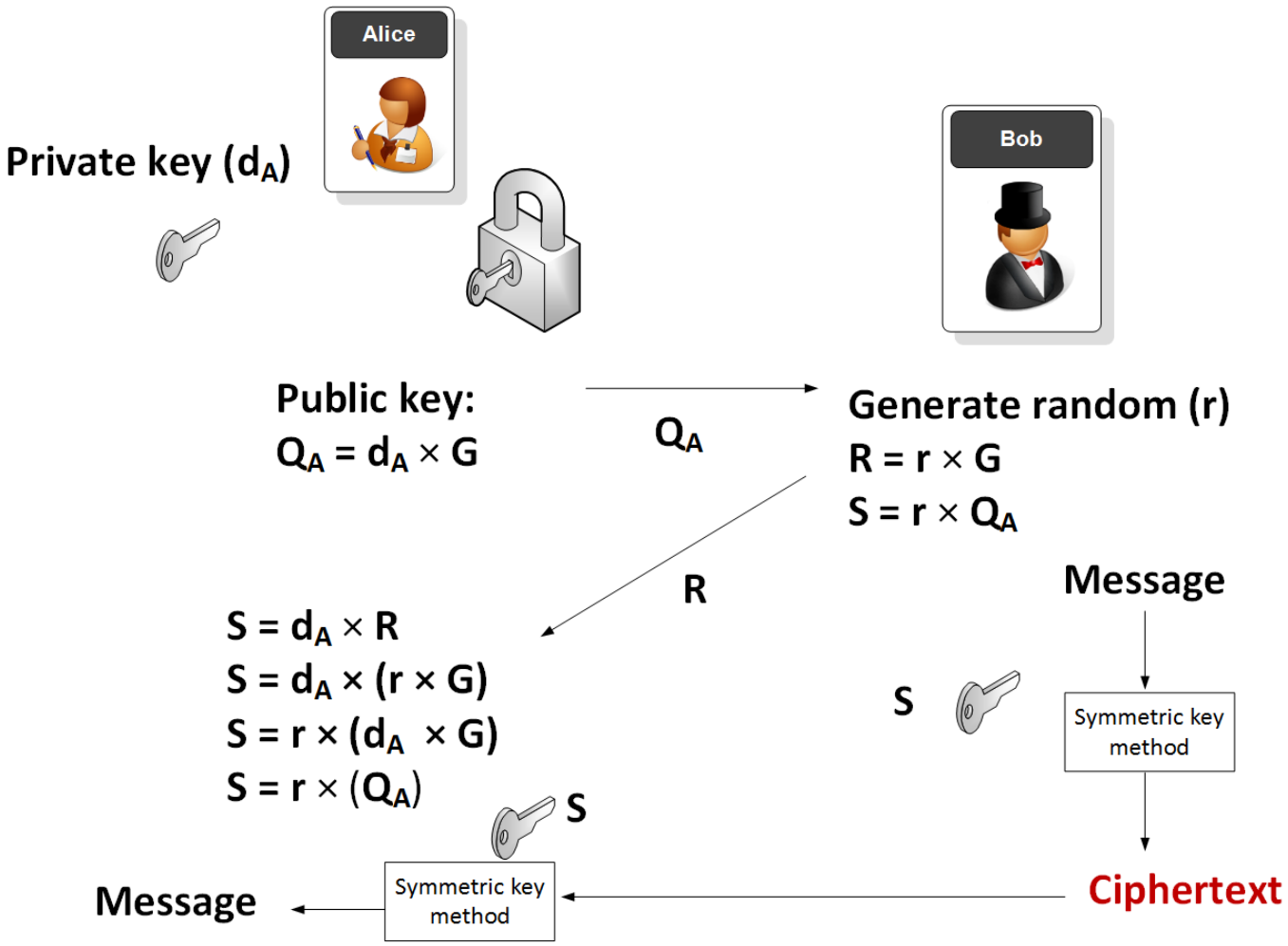

Figure: ECIES graphical explanation

ECIES combines a **K**ey **E**ncapsulation **M**echanism (**KEM**) with a **D**ata **E**ncapsulation **M**echanism (**DEM**). The system independently derives a bulk encryption key and a MAC key from a common secret. Data is first encrypted under a symmetric cipher, and then the cipher text is MAC'd under an authentication scheme.

Finally, the common secret is encrypted under the public part of a public/private key pair. The output of the encryption function is the tuple $\{K, C, T\}$, where K is the encrypted common secret, C is the ciphertext, and T is the authentication tag.

Elliptic Curve Integrated Encryption Scheme

Advantages

- ECIES and ACE use the first coordinate of a point of the curve generated during the calculations (instead of both coordinates) as an input parameter to the key derivation function previously mentioned, while PSEC requires to use both coordinates.
- PSEC uses twice a key derivation function in order to obtain a pair of MAC and symmetric encryption keys, whilst ACE and ECIES use such a function only once.
- More secure than RSA (256bits vs 3072bits).

Elliptic Curve Integrated Encryption Scheme

Drawbacks

- Slow to encrypt large plaintexts.
- Mathematics are more complex (engineers feels to understand better RSA).

Índex

Signature schemes
Explanation

Signatures schemes are the digital counterparts to handwritten signatures. They can be used to provide *data origin authentication*, *data integrity*, and *non-repudiation*. Signature schemes are commonly used by trusted certification authorities to sign certificates that bind together an entity and its public key.

Signature schemes

Steps

1. A *domain parameter generation algorithm* that generates a set D of domain parameters.
2. A *key generation algorithm* that takes as input a set D of domain parameters and generates key pairs (Q, d).
3. A *signature generation algorithm* that takes as input a set of domain parameters D, a private key d, and a message m, and produces a signature Σ.
4. A *signature verification algorithm* that takes as input the domain parameters D, a public key Q, a message m, and a purported signature Σ, and accepts or rejects the signature.

Elliptic Curve Digital Signature Algorithm

Explanation

The **E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm (**ECDSA**) is the elliptic curve analog of the **D**igital **S**ignature **A**lgorithm (**DSA**). Is the most widely standardized elliptic curve-based signature scheme, appearing in the *ANSI X9.62, FIPS 186-2, IEEE 13632000* and *ISO/IEC 15946-2* standards.

With ECDSA, Alice will sign a message with her private key, and then Bob will use her public key to verify that she signed the message:

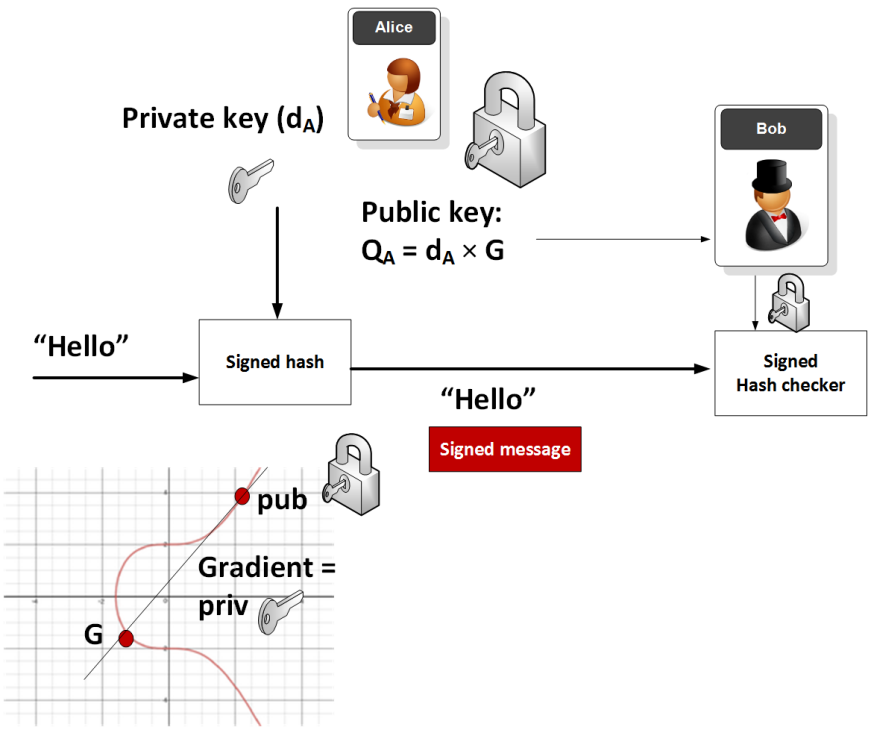

Figure: ECDSA graphical explanation

Alice

Alice signs the message with the following:

1. Create a hash of the message $e = HASH(m)$.
2. Let z be the L_n be the leftmost bits of e, L_n has a bit length of the group order n.
3. Create a random number k which is between 1 and n-1.
4. Calculate a point on the curve as $(x1,y1) = k \times G$.
5. Calculate $r = x_1$ mod n. If r = 0, go back to *Step 3*.
6. Calculate $s = k$ - $1(z + rd_A)$mod n. If s = 0, go back to *Step 3*.
7. The signature is the pair (r,s).

Bob

Bob will check with:

1. Create a hash of the message $e = HASH(m)$.
2. Let z be the L_n leftmost bits of e.
3. Calculate $c = s$ - $1 \bmod n$
4. Calculate $u_1 = z * cmodn$ and $u_2 = r * cmodn$.
5. Calculate the curve point $(x_1, y_1) = u_1 * G + u_2 * Q_A$. If $(x_1, y_1) = O$ then the signature is invalid.
6. The signature is valid if r$\equiv x_1 (\text{mod n})$, invalid otherwise.

Elliptic Curve Digital Signature Algorithm

Advantages

- As with ellipticcurve cryptography in general, the bit size of the public key believed to be needed for *ECDSA* is about twice the size of the security level, in bits.
- Using n-bit ECDSA, a signature has a size of 2*n.
- We can use a lot of hash functions to perform our strength security attack.

Elliptic Curve Digital Signature Algorithm

Drawbacks

- The difficulty of properly implementing the standard, its slowness, and design flaws which reduce security in insufficiently defensive implementations of the Dual EC random number generator.
- The security depends undirectly on the integrity of the NIST curves used to sign.

Índex

Key establishment
Explanation

The purpose of a key establishment protocol is to provide two or more entities communicating over an open network with a shared secret key. The key may then be used in a symmetric-key protocol to achieve some cryptographic goal such as confidentiality or data integrity.

Elliptic Curve Diffie-Hellman

Explanation

Elliptic **C**urve **D**iffie-**H**ellman (**ECDH**) is an anonymous key agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel.

This shared secret may be directly used as a key, or to derive another key. The key, or the derived key, can then be used to encrypt subsequent communications using a symmetric-key cipher. It is a variant of the *Diffie-Hellman* protocol using elliptic curve cryptography.

ECDH

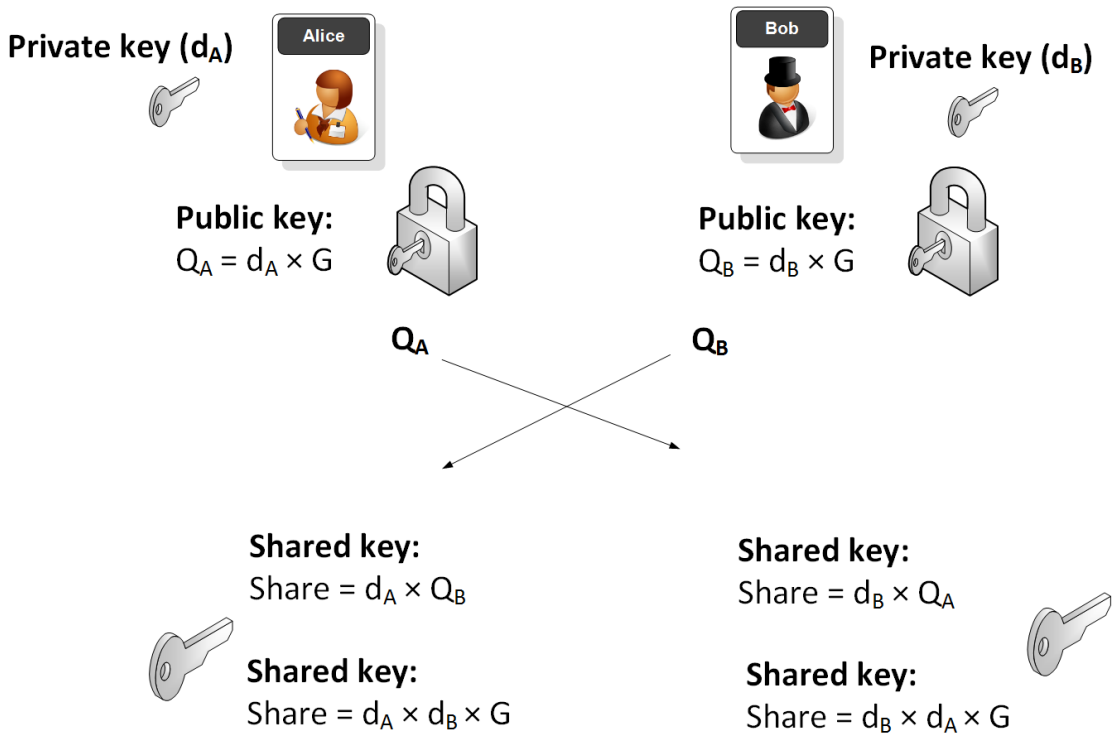

Figure: ECDH graphical explanation

1. First, Alice and Bob generate their own private and public keys. We have the private key d_A and the public key $Q_A = d_A G$ for Alice, and the keys d_B and $Q_B = d_B G$ for Bob. So Alice's key pair be (d_A, Q_A), and Bob's key pair be (d_B, Q_B). Each party must know the other party's public key prior to execution of the protocol. Note that both Alice and Bob are using the same domain parameters: the same base point G on the same elliptic curve on the same finite field

2. Alice and Bob exchange their public keys Q_A and Q_B over an insecure channel. The Man In the Middle would intercept Q_A and Q_B, but won't be able to find out neither d_A nor d_B without solving the discrete logarithm problem.

3. Alice calculates $S = d_A Q_B$ (using her own private key and Bob's public key), and Bob calculates $S = d_A Q_A$ (using his own private key and Alice's public key). Note that S is the same for both Alice and Bob, in fact:

$$S = d_A Q_B = d_A(d_B G) = d_B(d_A G) = d_B Q_A \qquad (6)$$

Elliptic Curve Diffie-Hellman

Advantages

- One strong are that ECDH crypto system can be successfully adapted to different data security solutions.
- Is one of the most used crypto systems that are standardized and patent free. So, there is free to use.
- ECDH key exchange is very frequent used in WSNs (Wireless Sensor Networks) because it has a perfect resilience to node capture, excellent scalability, and low memory as well as communication over-head.

Elliptic Curve Diffie-Hellman

Drawbacks

- Key exchange protocol does not protect against active attacks. Alice must use some method to ensure that Q_A originated with the intended communicant Bob, rather than an attacker, and Bob must do the same with Q_B.
- If we have an elliptic curve

$$y^2 = x^3 + a * x + b(modp) \tag{7}$$

The elliptic curve group operation does not explicitly incorporate the parameter b from the curve equation. This opens the possibility that a malicious attacker could learn information about an *ECDH* private key by submitting a bogus public key.

Elliptic Curve Diffie-Hellman

Drawbacks

- However the big drawback of *ECDH* in **WSN**s (**W**ireless **S**ensor **N**etworks) is the highly computation-intensive nature of its underlying cryptographic operations, causing long execution times and high energy consumption.

Índex

Safe Curves

https://safecurves.cr.yp.to/

References

- https://asecuritysite.com/encryption/ecdsa
- https://asecuritysite.com/encryption/ecdh
- https://safecurves.cr.yp.to/
- https://medium.com/asecuritysite-when-bob-met-alice/generating-an-encryption-key-without-a-pass-phrase
- https://asecuritysite.com/encryption/ecc2