Cryptography Workshop

Part 10: Elliptic Curve Cryptography

Prof. Dr. Gerd Beuster

Summer Semester 2015
(Created: 09.06.2015)

Contents

1 Introduction

Textbook

Today's lecture is based on Chapter 10 (Other Public-Key Cryptosystems) of Stallings' book. Many of the figures come straight from the book.

	Asymmetric Encryption	Digital Signatures	Key Exchange
Prime Factorization	RSA	RSA	
Discrete Logarithm	*Elgamal*	*Elgamal*	DH

Table 1: Keysharing and Public Key Crypto

Elgamal
Asymmetric encryption & digital signatures based on DLP.

Elliptic Curve Cryptography
Use elliptic curve arithmetic instead of modular arithmetic.

- Straightforward for DLP
- Not-so-straightforward for PFP

2 Elgamal

Elgamal
Asymmetric encryption scheme based on DLP.

Let ...

q	Modulus; a large prime (public)
α	Generator of multiplicative group q (public)
X_A	Alice's secret
$Y_A = \alpha^{X_A} \pmod q$	Published by Alice

Elgamal encryption
Let M be the message. Generate random key k. The encrypted message is

$$(C_1; C_2)$$

with

$$\begin{aligned} C_1 &= \alpha^k && \pmod q \\ C_2 &= (Y_A)^k \cdot M && \pmod q \end{aligned}$$

Elgamal decryption
Alice must calculate $(Y_A)^k$ in order to decrypt the message. Since $Y_A = \alpha^{X_A}$,

$(Y_A)^k = \left(\alpha^{X_A}\right)^k$. Since Alice gets the "hint" $C_1 = \alpha^k$, she can calculate

$$(C_1)^{X_A} = \left(\alpha^k\right)^{X_A} = \left(\alpha^{X_A}\right)^k = (Y_A)^k$$

Now she calculates the inverse element $\left((Y_A)^k\right)^{-1}$. She decrypts the message by calculating

$$C_2 \cdot \left((Y_A)^k\right)^{-1} = (Y_A)^k \cdot M \cdot \left((Y_A)^k\right)^{-1} = M \pmod{q}$$

Example

This example is given in [Stal13, p. 314].

Agree on parameters

In this example, we choose $q = 19$ and $\alpha = 10$ as a primitve root of $GF(19)$.

Alice generates her public and private keys

Alice picks her private $X_A = 5$.

She calculates her public key as

$$Y_A = \alpha^{X_A} = 10^5 = 3 \pmod{19}$$

Encryption

Bobs wants to encrypt $M = 17$. He generates a random number $k = 6$ and calculates

$$\begin{aligned} C_1 &= \alpha^k = 10^6 = 11 && \pmod{19} \\ C_2 &= (Y_A)^k \cdot M = 3^6 \cdot 17 = 5 && \pmod{19} \end{aligned}$$

He sends $(C_1; C_2) = (11; 19)$ to Alice.

Decryption

Alice calculates

$$(Y_A)^k = (C_1)^{X_a} = 11^5 = 7 \pmod{19}$$

The inverse element is

$$\left((Y_A)^k\right)^{-1} = 7^{-1} \pmod{19} = 11$$

She recovers the plaintext by calculating

$$M = C_2 \cdot \left((Y_A)^k\right)^{-1} = 5 \cdot 11 = 17 \pmod{q}$$

The following explanation and example comes straight from [Stal13, p.418–420]. In order to understand the calculations, note that it can be proofed that for every message m and for all integers i, j:

- $\alpha^m = 1 \pmod{q}$ iff $m = 0 \pmod{q-1}$
- $\alpha^i = \alpha^j \pmod{q}$ iff $i = j \pmod{q-1}$

Elgamal Signatures
The Elgamal algorithm can also be used for signing messages.

Alice creates her public and private key as for encryption.

Private key: Random number X_A
Public key: $Y_A = \alpha^{X_A} \pmod{q}$

Signing a message
Alice chooses a random integer k with $1 \leq k \leq q-1$ and $\gcd(k, q-1) = 1$

She calculates $S_1 = \alpha^k \pmod{q}$. This is the same calculation as C_1 in the encryption scenario. Just like C_1, S_1 serves as a "hint" to the secret k.

She computes $k^{-1} \pmod{q-1}$ and $S_2 = k^{-1} \cdot (M - X_A \cdot S_1) \pmod{q-1}$.

$(S_1; S_2)$ is the signature of message M.

Checking the signature
Bob calculates $V_1 = \alpha^M \pmod{q}$ and $V_2 = (Y_A)^{S_1} \cdot (S_1)^{S_2} \pmod{q}$

The signature is valid if $V_1 = V_2$.

This works because:

	V_1	$= V_2$		
$\Leftrightarrow$	α^M	$= (Y_A)^{S_1} \cdot (S_1)^{S_2}$	$\pmod{q}$	substituting for V_1 and V_2
$\Leftrightarrow$	α^M	$= \alpha^{X_a \cdot S_1} \cdot \alpha^{k \cdot S_2}$	$\pmod{q}$	substituting for Y_A and S_1
$\Leftrightarrow$	$\alpha^{M - X_a \cdot S_1}$	$= \alpha^{k \cdot S_2}$	$\pmod{q}$	rearranging terms
$\Leftrightarrow$	$M - X_a \cdot S_1$	$= k \cdot S_2$	$\pmod{q-1}$	property of primitive roots
$\Leftrightarrow$	$M - X_a \cdot S_1$	$= k \cdot k^{-1} \cdot (M - X_A \cdot S_1)$	$\pmod{q-1}$	substituting for S_2

Example

Again, we use $GF(19)$ with $\alpha = 10$.

Alice's signature key

Alice picks $X_A = 16$ as her private signature key. She calculates her public signature key as $Y_A = \alpha^{X_A} = 10^{16} = 4 \pmod{19}$.

Alice signs message

She wants to sign $M = 14$. She picks random $k = 5$, which is relatively prime to $q - 1 = 18$. She calculates $k^{-1} = 5^{-1} = 11 \pmod{18}$ and then

$$\begin{array}{lll} S_1 & = \alpha^k = 10^5 = 3 & \pmod{19} \\ S_2 & = k^{-1} \cdot (M - X_A \cdot S_1) & \pmod{18} \\ & = 11 \cdot (14 - 16 \cdot 3) & \pmod{18} \\ & = 4 & \end{array}$$

The signature is $(S_1; S_2) = (3; 4)$.

Note that S_1 is calculated modulo q and S_2 modulo $q - 1$.

Checking the signature

Bob calculates

$$\begin{array}{lll} V_1 & = \alpha^M = 10^{14} = 16 & \pmod{19} \\ V_2 & = (Y_a)_1^S \cdot (S_1)^{S_2} = 4^3 \cdot 3^4 = 16 & \pmod{19} \end{array}$$

Since $V_1 = V_2$, the signature is valid.

3 Elliptic Curves

So far,
we used finite field arithmetic modulo a prime p or p^n.

Now
We use elliptic curves instead.

Elliptic Curves
form an *algebraic group* with a suitable parameter.

An elliptic curve satifies the formula

$$y^2 = x^3 + a \cdot x + b$$

Coefficients a and b are fixed.

Figure 1 shows two elliptic cuves curves.

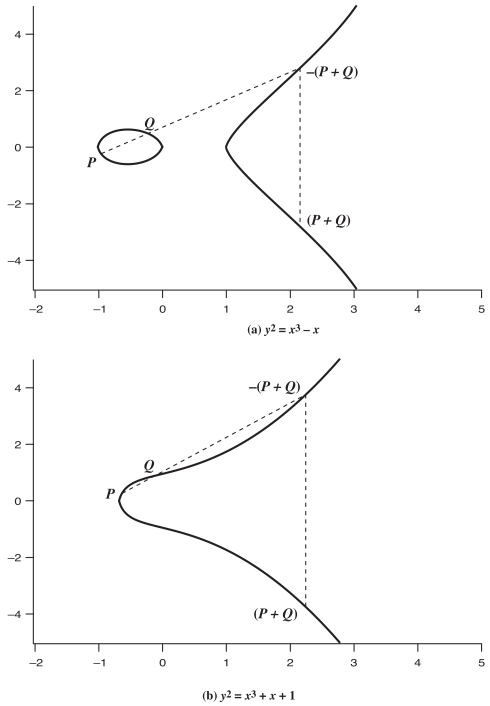

Figure 1: Example of Elliptic Curves

Definition
An elliptic curve is defined by $(a; b)$. We write $E(a; b)$.

Points
on the eliptic curves are given by their $(x; y)$ coordinates.

Arithmetic
In order to build a group, we have to define *addition*.

We also need a neutral element, called 0 (point at infinity, zero point).

Negation
A point P is negated by negating its y coordinate:

$$P = (x; y)$$
$$-P = (x; -y)$$

Addition
Draw straight line through P and Q.

Find third point of intersection with curve.

This point is $-(P + Q)$.

See Figure 1

Double point P
Draw tangent line of point P.

The point where it intersects the curve is $-(P + P)$.

Formulas
The formulas for the addition operation are omitted here. If you are interested, see for example [Stal13, p. 312]

Finite elliptic curves
can be defined over $\mathbb{Z}_p$ or $GF(p^n)$ with p a prime.

Cryptography
In cryptography, only curves over $\mathbb{Z}_p$ and $GF(2^n)$ are used.

Here
we only consider curves over $\mathbb{Z}_p$.

Curves over $\mathbb{Z}_p$

Same calculations as over infinite curves, but with integers modulo p**.**

$$y^2 = x^3 + a \cdot x + b \pmod{p}$$

Example: Curve $E_{23}(1; 1)$

$$y^2 = x^3 + x + 1 \pmod{23}$$

Table 2 lists the points on the curve, and Figure 2 shows the graph of the curve. As you can see, the operations on a discrete elliptic curve do not have an easy geometric interpretation.

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

Table 2: Points on Elliptic Curve $E_{23}(1; 1)$

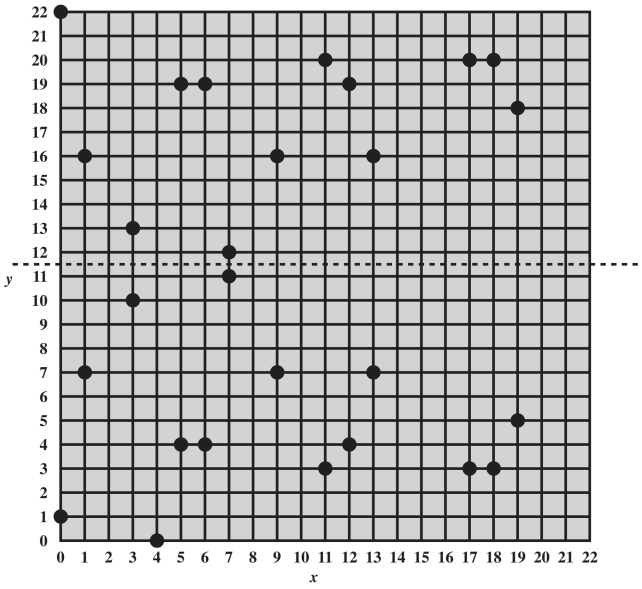

Figure 2: The Elliptic Curve $E_{23}(1; 1)$

$E_p(a, b)$ with elliptic curve addition defines a finite abelian group if

$$4 \cdot a^3 + 27 \cdot b^2 \neq 0 \pmod{p}$$

Note that $E_p(a, b)$ defines a group, not a ring or field.
$\Longrightarrow$ There is no multiplication in the true sense.

E.g.
We cannot compute $(1; 16) \cdot (17; 20)$

But we can define multiplication with an integer as repeated addition.

$$\begin{aligned} 5 \cdot (1; 16) &= (1; 16) + (1; 16) + (1; 16) + (1; 16) + (1; 16) \\ &= (0; 22) \end{aligned}$$

4 Elliptic Curve Cryptography

Adapting DH and Elgamal for EC-Cryptography is straightforward.
Use *EC multiplication* instead of *integer exponentiation* as basic operation.

4.1 Elliptic Curve Diffie-Hellman

Let ...

$a; b; p$	Define the finite field $E_p(a; b)$
G	Be a point of large order n on the curve
n_A	Alice's secret with $n_A < n$
$P_A = G \cdot n_A$	Published by Alice
n_B	Bob's secret with $n_A < n$
$P_B = G \cdot n_B$	Published by Bob

Alice computes the shared secret as

$$P_B \cdot n_A = G \cdot n_B \cdot n_A$$

Bobs computes the shared secret as

$$P_A \cdot n_B = G \cdot n_A \cdot n_B$$

4.2 Elliptic Curve Elgamal

Let ...

$a; b; p$	Define the finite field $E_p(a; b)$
G	Be a point of large order n on the curve
n_A	Alice's secret with $n_A < n$
$P_A = G \cdot n_A$	Published by Alice

Furthermore, let

M	Bob's message to Alice
k	An random number choosen by Bob

Encryption

Bob calculates

$$C_1 = k \cdot G$$
$$C_2 = M + k \cdot P_A$$

He sends $(C_1; C_2)$ to Alice.

Just like in the normal Elgamal scheme, C_1 is the "hint" needed by Alice to decrypt the message.

Decryption

Alice calculates

$$\begin{aligned} X &= n_A \cdot C_1 = n_A \cdot k \cdot G \\ C_2 - X &= M + k \cdot P_A - X \\ &= M + k \cdot G \cdot n_A - n_a \cdot k \cdot G \\ &= M \end{aligned}$$

Example: Let

$E_{257}(0; -4)$	be the elliptic curve
$G = (2; 2)$	the generator
$n_a = 101$	Alice's private key
$P_A = n_a \cdot G = (197; 167)$	Alice's public key

Encryption

Bob wants to encrypt $M = (112; 26)$. He picks random number $k = 41$ and calculates

$$C_1 = k \cdot G = 41 \cdot (197; 167) = (136; 128)$$
$$C_2 = M + k \cdot P_A = (112; 26) + 41 \cdot (197; 167) = (246; 174)$$

He sends $((136; 128); (246; 174))$ to Alice.

Decryption

Alice calculates

$$\begin{aligned} X &= n_A \cdot C_1 = 101 * (136; 128) = (68; 84) \\ C_2 - X &= (246; 174) - (68; 84) \\ &= (112; 26) \\ &= M \end{aligned}$$

Security

ECDLP is considered harder than DLP and prime factorization.Table 3 shows the key bit lengths required by different crypto algorithms in order to achieve the same security level.

Bits of security	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
80	2TDEA	L = 1024 N = 160	k = 1024	f = 160-223
112	3TDEA	L = 2048 N = 224	k = 2048	f = 224-255
128	AES-128	L = 3072 N = 256	k = 3072	f = 256-383
192	AES-192	L = 7680 N = 384	k = 7680	f = 384-511
256	AES-256	L = 15360 N = 512	k = 15360	f = 512+

Table 3: Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis [Ni12, p. 64]

References

[Ni12] National Institute of Standards and Technology (NIST), NIST Special Publication 800-57: Recommendation for Key Management – Part 1: General, Revision 3, NIST, Gaithersburg, MD 20899-8930, USA, 2012, http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf

[Stal13] William Stallings, Cryptography and Network Security, Pearson, 6th ed., 2013