Bilinear Pairings

High Level Implementation

Luis J. Dominguez Perez

CRYPTO-CO, Julio 5 de 2016

Contenido, sección 1

Introduction

Pairings in cryptography

Pairing protocols

Appendix

Group

A group $< G, \circ >$ is a non-empty set G together with a binary operation $\circ$ such that:

- *It is closed*
- *it is associative*
- *has an identity and inverse element*

A group G is **Abelian** (or **commutative**) if $a \circ b = b \circ a$, $\forall a, b \in G$.

A group is **finite** if G has a finite number of elements. This is called the **order** of G and denoted as $|G|$.

Finite field

A field F is a group with $+$, $\times$ operations as $(F, +)$ and $(F \backslash \{0\}, \times)$ which also satisfies:

- *Additive identity and inverse*
- *Multiplicative identity and inverse*
- *Commutative*

i.e. the set of integers modulo p-prime, also denoted as $\mathbb{F}_p$, is a finite field.

Elliptic curves over finite fields

Let p-prime > 3. The elliptic curve

$$y^2 = x^3 + ax + b, \qquad \text{over } \mathbb{F}_p$$

denoted by $E(\mathbb{F}_p)$, is the set of solutions $x, y \in \mathbb{F}_p$ satisfying

$$y^2 \equiv x^3 + ax + b \bmod p$$

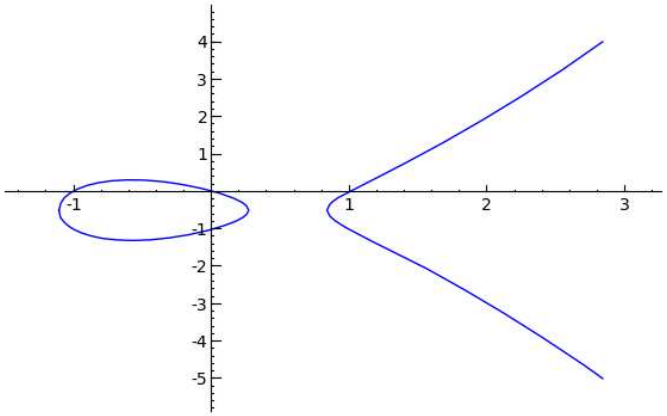
where $a, b \in \mathbb{F}_p$ and

$$4a^3 + 27b^2 \neq 0 \bmod p$$

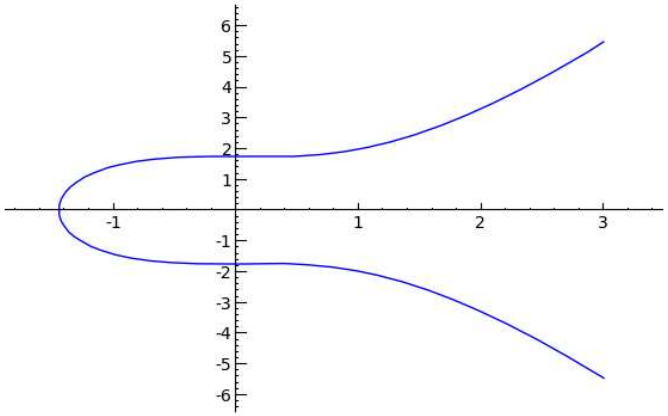
together with the point $\mathcal{O}$

The **order**, or number of points on $E(\mathbb{F}_p)$ is denoted as $\#E(\mathbb{F}_p) = p + 1 \pm t$ and $t \leq 2\sqrt{p}$.

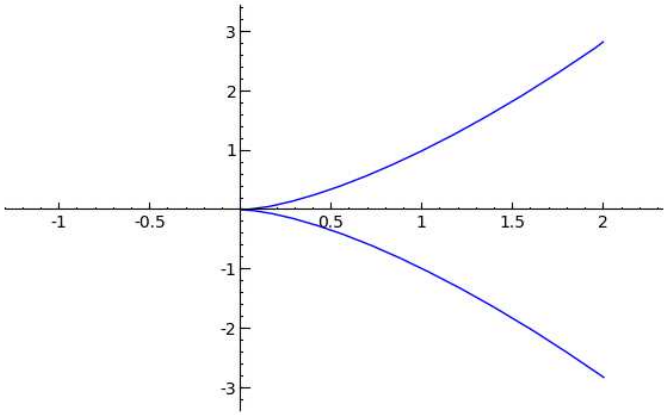
Types of elliptic curves (over $\mathbb{C}$)



With 3 distinct real roots



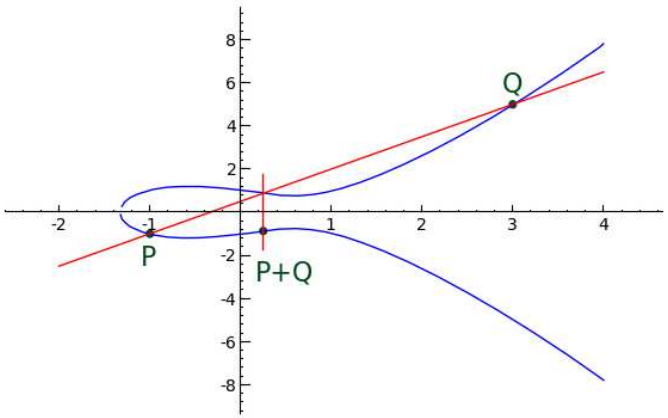
With 1 real and 2 complex roots



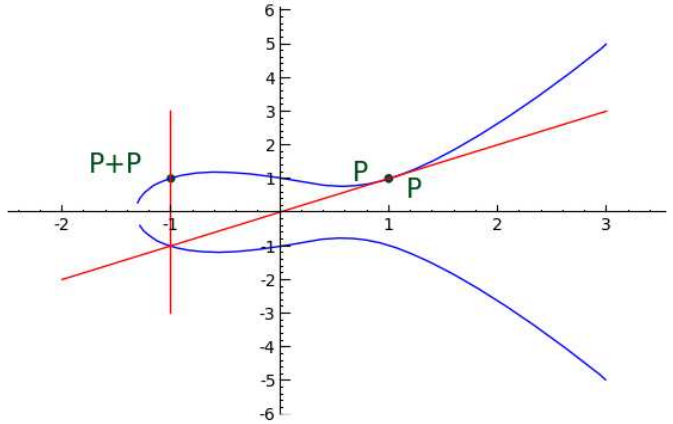
With a triple real root

The group law on EC

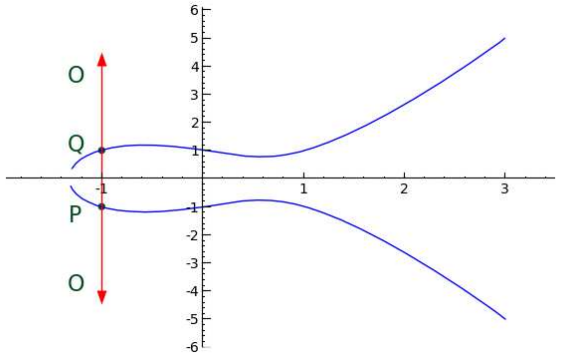
Suppose $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ with $P, Q \in E(\mathbb{F}_p)$ $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2, y_3 = \lambda(x_1 - x_3) - y_1$



$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$



$$\lambda = \frac{3x_1^2 + a}{2y_1}$$



$$P + Q = \mathcal{O} \text{ or } Q = -P$$

Families of elliptic curves

Let the subgroup size be the large prime number $r \mid \#E$. Let k be the **embedding degree**, $r \mid (p^k - 1)$.

A pairing-friendly elliptic curve has a **small embedding degree** and a **large subgroup size**[1].

Random curves cannot meet these requirement. A family of pairing-friendly elliptic curves use polynomials as parameters and suit a **required security level**.

[1] $k < 50$, r 160-bits

KSS curves, k=18

Kachisa et al. [2008] presented a new method for constructing pairing-friendly elliptic curves.

The parameters of these types of curves are:

- $t(x) = (x^4 + 16x + 7)/7$
- $p(x) = (x^8 + 5x^7 + 7x^6 + 37x^5 + 188x^4 + 259x^3 + 343x^2 + 1763x + 2401)/21$
- $r(x) = (x^6 + 37x^3 + 343)/343$
- Let $\rho \approx \frac{\deg(p(x))}{\deg(r(x))} = 4/3$.

$p(x)$ and $r(x)$ represent primes and $t(x)$ represents integers when $x \equiv 14 \bmod 42$

More curves

Other families of pairing-friendly elliptic curves with different embedding degrees:

- MNT curves
- Freeman curves
- BN curves
- KSS curves

For an extended description of these and other families of pairing-friendly elliptic curves, refer to Freeman et al. [2006]

Contenido, sección 2

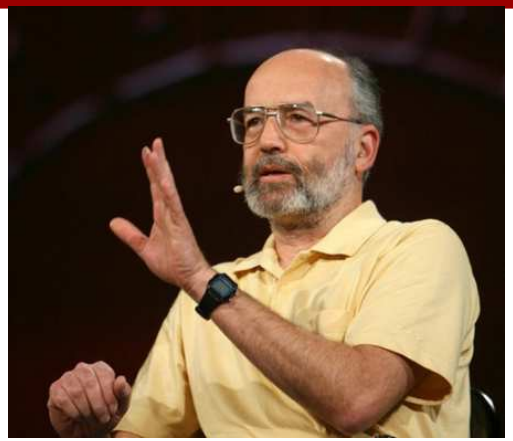
Introduction

Pairings in cryptography

Pairing protocols

Appendix

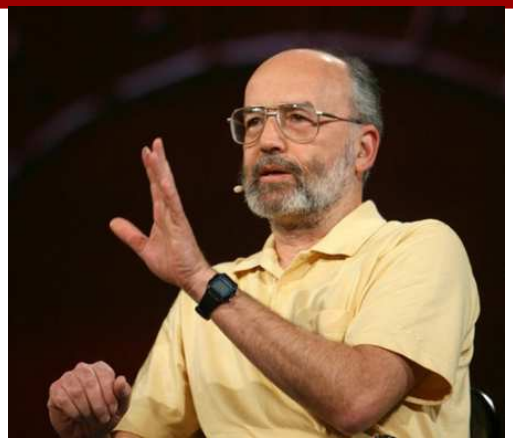
How does it started?



In 1984, Shamir posed a challenge:

"create a cryptographic system that permits any two users to comunicate securely and to verify each other's signatures without exchanging private or public keys, without keeping key directories, and without using the services of a third party."

How does it started?



In 1984, Shamir posed a challenge:

"create a cryptographic system that permits any two users to comunicate securely and to verify each other's signatures without exchanging private or public keys, without keeping key directories, and without using the services of a third party."

This sounds impossible!

How does it started? 2

However, in 2001, Boneh and Franklin, solved this challenge using cryptographic pairings. They presented what it is now called Identity-Based Encryption.

How does it started? 2

However, in 2001, Boneh and Franklin, solved this challenge using cryptographic pairings. They presented what it is now called Identity-Based Encryption.

... also, in 2000, Antoine Joux presented a breaking-through paper involving pairings, but we are focusing in this talk on Identity-Based Encryption.

Pairing-Based Cryptography

Identity-Based Encryption is a type of the Pairing-Based Encryption, this is, we use some cryptographic function called the pairing.

In essence, a cryptographic pairing is a particular function of groups over elliptic curves.

$$\langle \cdot, \cdot \rangle : M \times M \longrightarrow R$$

The bilinear pairing can be used as a primitive to build cryptosystems with certain functionality. Examples of use:

The bilinear pairing can be used as a primitive to build cryptosystems with certain functionality. Examples of use:

- Short signatures schemes,

The bilinear pairing can be used as a primitive to build cryptosystems with certain functionality. Examples of use:

- Short signatures schemes,
- Identity-Based Encryption,

The bilinear pairing can be used as a primitive to build cryptosystems with certain functionality. Examples of use:

- Short signatures schemes,
- Identity-Based Encryption,
- Attribute-Based Encryption,

The bilinear pairing can be used as a primitive to build cryptosystems with certain functionality. Examples of use:

- Short signatures schemes,
- Identity-Based Encryption,
- Attribute-Based Encryption,
- and other protocols already deployed.

Some protocols are impossible with currently deployed technology, in other cases, they are faster.

Example of PBC

Identity-Based Encryption case:

- Enables any pair of users to communicate securely and to verify each others' signatures **without exchanging** private or public keys;
- Needs **no key server repositories**;
- Requires a trusted server for key generation **only**.
- **No certificate required** to bind the public key to the identity.

Implementation issues...

Pairing-Based Cryptography has become relevant in industry.

Although there are plenty of applications, however efficiently implementing the pairings function is often difficult as it requires more knowledge than previous cryptographic primitives.

There are many implementation issues just with the primitive itself!

... implementation issues

- **Non-familiar** technology;
- Lack of **programming framework**;
- More **difficult to understand** compared to the already deployed technology;
- **Unavailability** of implementations with novel (faster) computing methods;
- Complex area.

Depending on the scenario, a developer must choose from a selection of parameters and apply the corresponding optimizations for efficiency...

What to do when... ?

- **bandwidth** use is expensive;
- **low memory** is available;
- a **slow** processor is used (old);
- a **small** processor (in bits) is the only option;
- we have a **Desktop** environment;
- we have a device with **multiprocessors**;
- a **higher security** is required;

Some basic operations that are cheap in some environments are expensive in others!

Protocol primitives

The operations involved in a Pairing-Based protocol are:

- The pairing function
- Elliptic Curve point addition and point doubling
- Scalar-point multiplication
- exponentiation
- hash onto a curve
- hash into a subgroup
- matrix conversion
- boolean function analysis. . .

Many more!

Some background

Let do a bit of maths...

Scalar-point multiplication

Let P be a point in a curve E and $n \in \mathbb{Z}$, $n \geq 0$. Define $[n]P = P + P + \cdots + P$. The order of the point P is the smallest n such that $[n]P = \mathcal{O}$.

Denote $< P >$ the group generated by P. In other words,

$$< P >= \{\mathcal{O}, P, P+P, P+P+P, \ldots\}$$

Let $Q \in < P >$. **Given Q, find n such that $Q = [n]P$ is hard.**

Applying the algorithm

The traditional method for computing the scalar-point multiplication is the Double-and-Add method.

Algorithm 1 Traditional scalar-point multiplication

Require: Positive integer k in base 2 representation, a point P.
Ensure: $[k]P$

1: $Q \leftarrow 0$
2: **for** $i = l - 1$ downto 0 **do**
3: $\quad Q \leftarrow [2]Q$
4: $\quad$ **if** $k_i = 1$ **then**
5: $\quad\quad Q \leftarrow Q + P$
6: $\quad$ **end if**
7: **end for**
8: **return** Q

Speeding up

Generic method to speed up the exponentiation in this context:

- Precomputation
- Addition chains whenever the scalar is known
- Windowing techniques
- Simultaneous multiple exponentiation techniques.

Replacing the binary representation of the scalar into one with fewer non-zero terms.

Speeding up II

Curve specific methods:

- A field defined with a (pseudo-)Mersenne prime.
- Field construction using small irreducible polynomials
- Point representation with fast arithmetic
- EC with special properties.

Pairing definition

A **pairing** is a map: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

These groups are finite and cyclic. $\mathbb{G}_1$ and $\mathbb{G}_2$ are additively-written and at least one is of prime order r. $\mathbb{G}_T$, is multiplicatively-written and of order r.

Properties:

- *Bilinearity*
- *Non-degeneracy*
- *Efficiently computable*

Pairing properties

Properties:

- *Bilinearity*

$$e(P + P', Q) = e(P, Q) \times e(P', Q)$$

$$e(P, Q + Q') = e(P, Q) \times e(P, Q')$$

Pairing properties

Properties:

- *Bilinearity*

  $$e(P + P', Q) = e(P, Q) \times e(P', Q)$$

  $$e(P, Q + Q') = e(P, Q) \times e(P, Q')$$

- *Non-degeneracy*

  $\forall P \in \mathbb{G}_1, P \neq \mathcal{O}$: $\exists Q \in \mathbb{G}_2$ **s.t.** $e(P, Q) \neq 1$

  $\forall Q \in \mathbb{G}_2, Q \neq \mathcal{O}$: $\exists P \in \mathbb{G}_1$ **s.t.** $e(P, Q) \neq 1$ $\quad$ $e(P, Q) \neq 1$

Pairing properties

Properties:

- *Bilinearity*

  $$e(P + P', Q) = e(P, Q) \times e(P', Q)$$

  $$e(P, Q + Q') = e(P, Q) \times e(P, Q')$$

- *Non-degeneracy*

  $\forall P \in \mathbb{G}_1,\ P \neq \mathcal{O}$: $\exists Q \in \mathbb{G}_2$ **s.t.** $e(P, Q) \neq 1$

  $\forall Q \in \mathbb{G}_2,\ Q \neq \mathcal{O}$: $\exists P \in \mathbb{G}_1$ **s.t.** $e(P, Q) \neq 1$ $\qquad$ $e(P, Q) \neq 1$

- *Efficiently computable*

(Ab)Using the pairing

The most important property of a pairing is:

$$e([a]Q, [b]P) = e([b]Q, [a]P) = e(Q, [ab]P) = e(Q, P)^{ab}$$

where $Q \in \mathbb{G}_2$, $P \in \mathbb{G}_1$, and the result is in $\mathbb{G}_T$.

In our context, the $\mathbb{G}_2$ group is larger than $\mathbb{G}_1$. The group $\mathbb{G}_T$ is also larger and has a different set of operations.

(Ab)Using the pairing II

- Since $\mathbb{G}_2$ is larger than $\mathbb{G}_1$, it is wise to exchange operations from one group to the other.

- $\mathbb{G}_T$ is significantly larger and has a different set of operations, we also try to avoid it, but we keep it handy, because...

(Ab)Using the pairing II

- Since $\mathbb{G}_2$ is larger than $\mathbb{G}_1$, it is wise to exchange operations from one group to the other.
- $\mathbb{G}_T$ is significantly larger and has a different set of operations, we also try to avoid it, but we keep it handy, because...
- An operation in $\mathbb{G}_T$ is cheaper than computing the pairing itself.

In short, we use the groups at will.

Using the pairing III

In each pairing-based protocol, we have to use the pairing function in a different way.

In some cases, we have to compute several-to-many pairings, in other cases, the pairings have the same parameters, in other cases, we have to add the results.

For this, we design the pairing function to be a:

- Multipairing
- Known-point pairing
- or we mix-up several parameters into a single pairing.

Contenido, sección 3

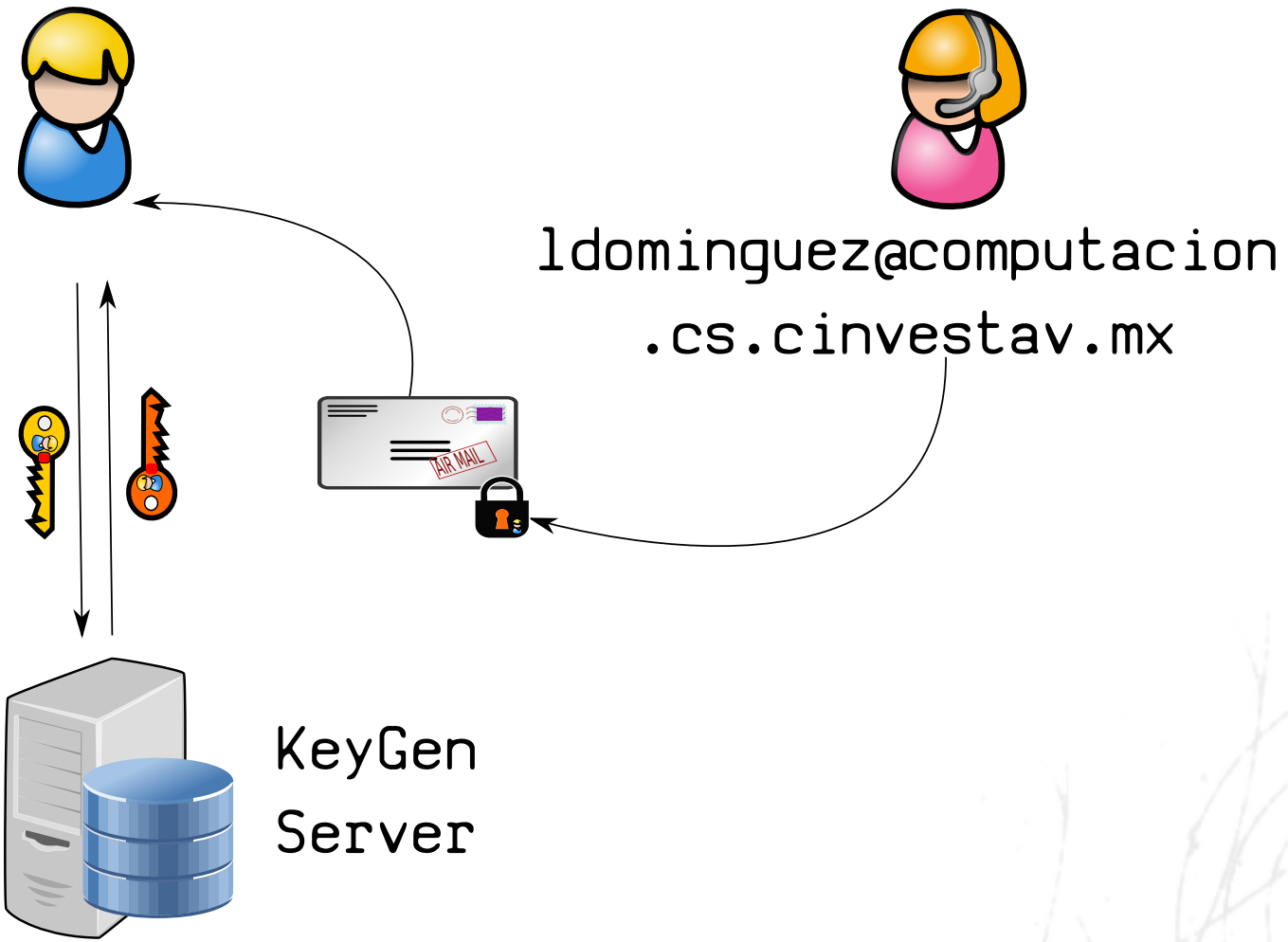
Introduction

Pairings in cryptography

Pairing protocols

Appendix

Encryption for an identity

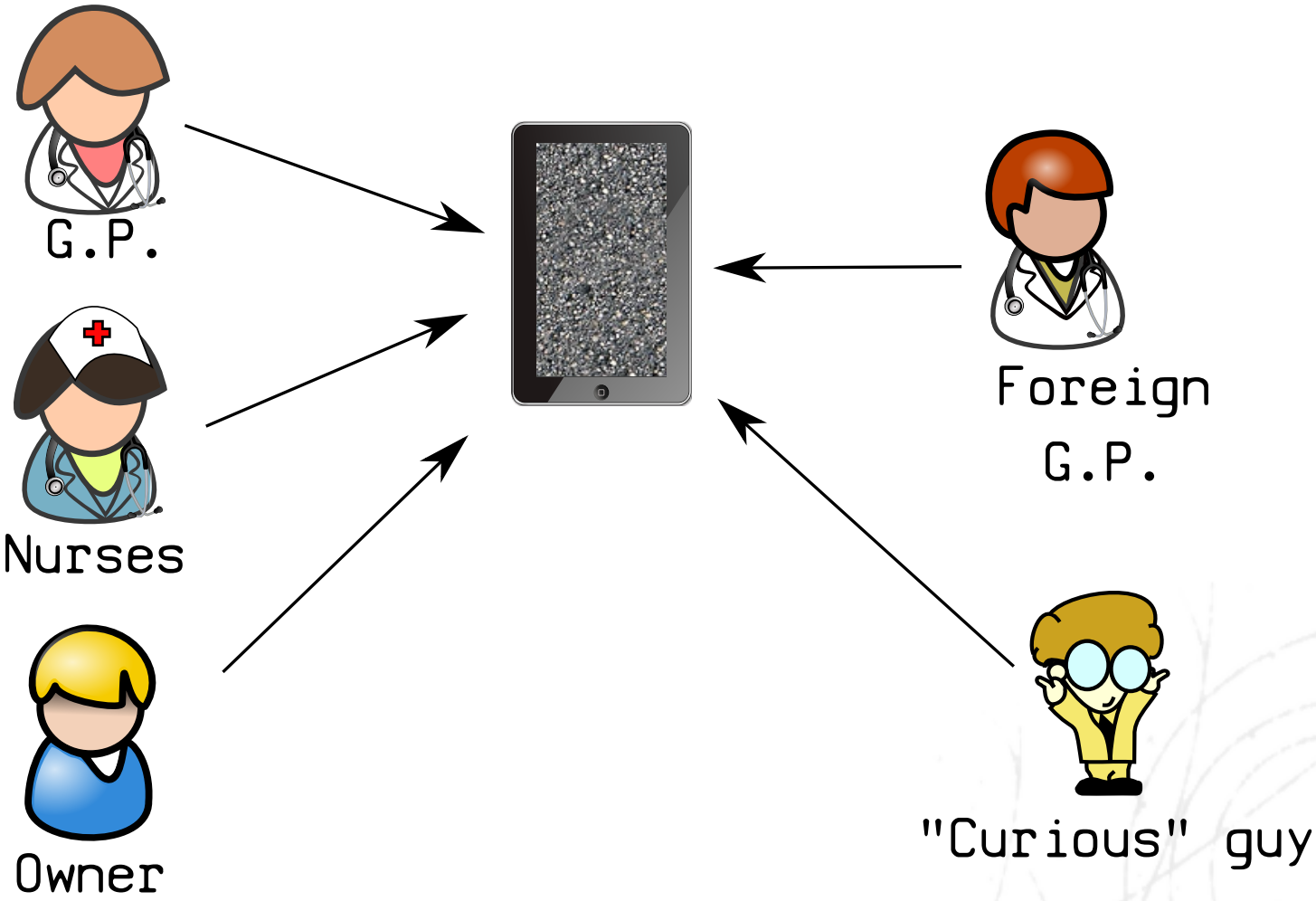


Attribute-Based Encryption

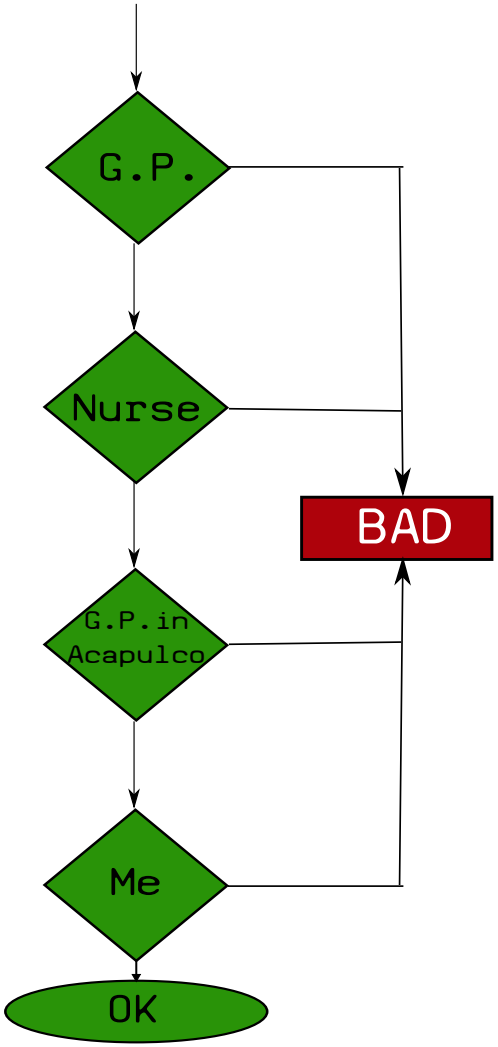
Fuzzy Identity-Based Encryption. Also known as Attribute-based encryption.

- An identity is a set of attributes
- An entity is valid if it presents a minimum number of attributes
- Better for sharing a small secret: a symmetric key.

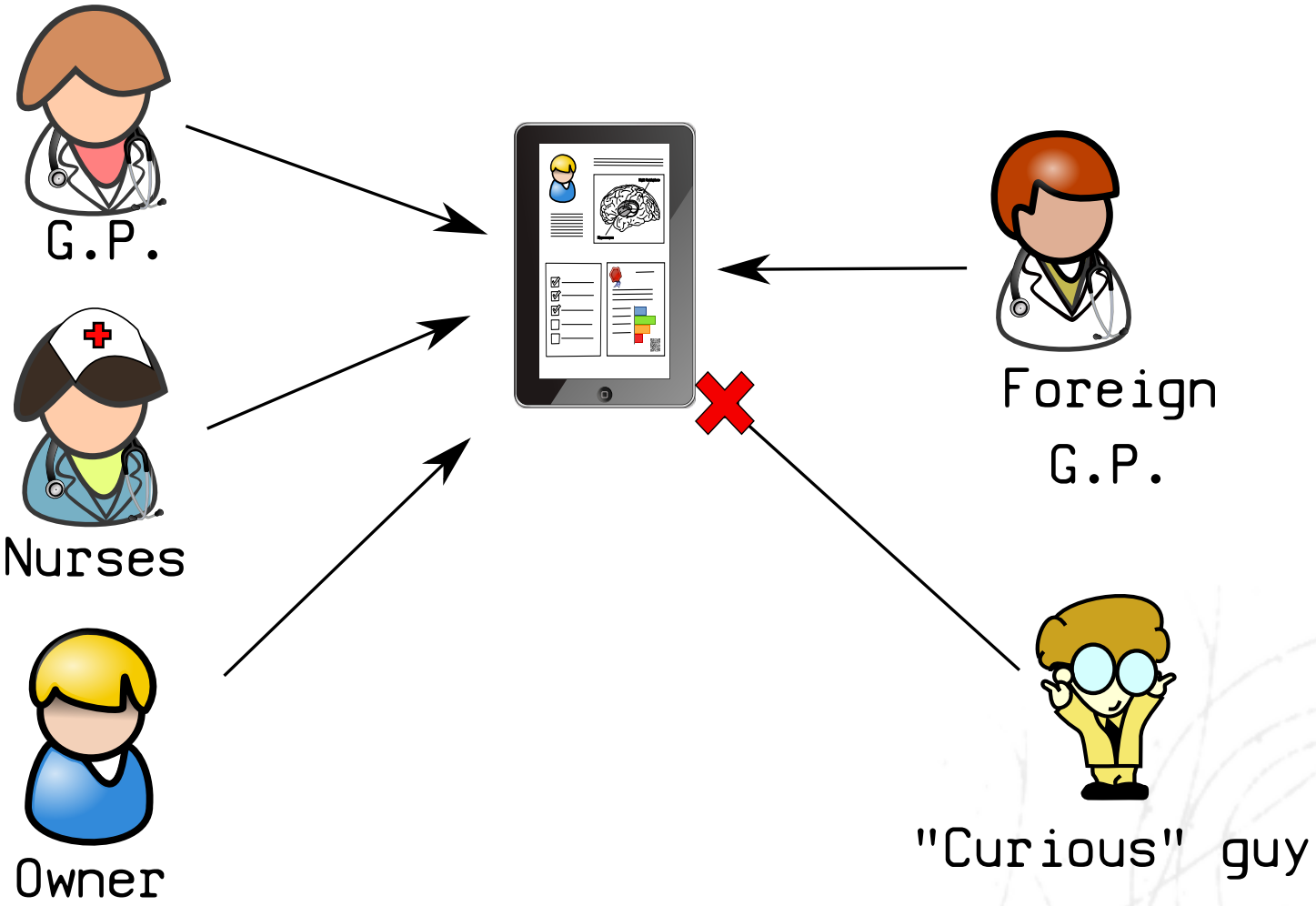
Attribute examples



Attribute examples



Attribute examples



Boneh's short signatures

Boneh's short signatures are based on the mathematical problem:

Given $(P, [a]P, Q, [b]Q)$, it is hard to decide if $a = b$

The computational variant of this hard problem is:

Given $(P, Q, [n]Q)$, compute $[n]P$

Boneh, Lynn and Shacham constructed a short signature scheme based on this problem as follows:

... the steps

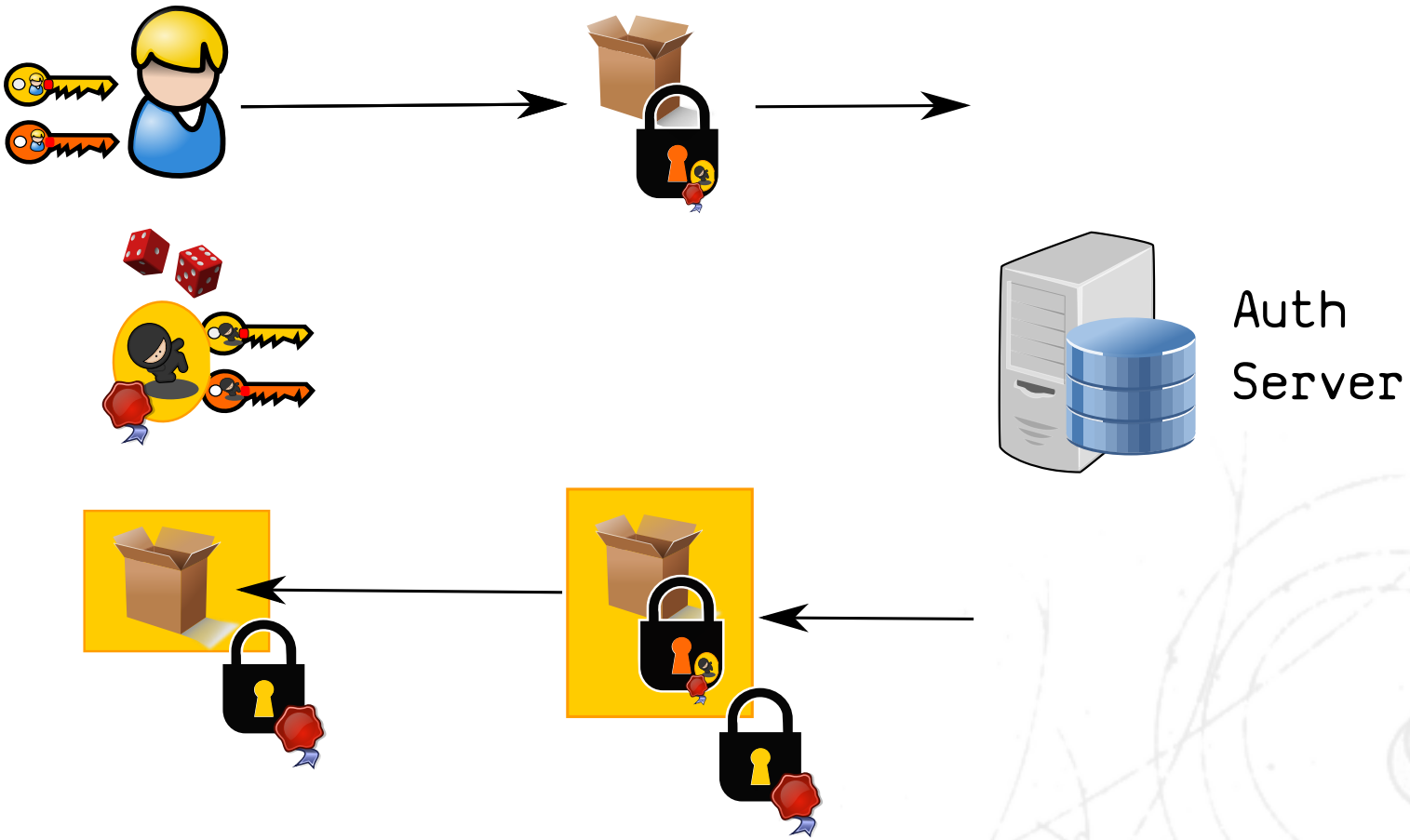
Key generation. Choose $n \in_R \mathbb{Z}_r$, set $R \leftarrow [n]Q$. The public key is: Q, R. The secret key is n

Sign. Map to a point the message to sign as P_M, set $S_M \leftarrow [n]P_M$. The signature is the x-coordinate of S_M.

Verify. Given the x-coordinate of S_M, find $\pm S$. Decide: $e(Q, S) \stackrel{?}{=} e(R, h(M))$

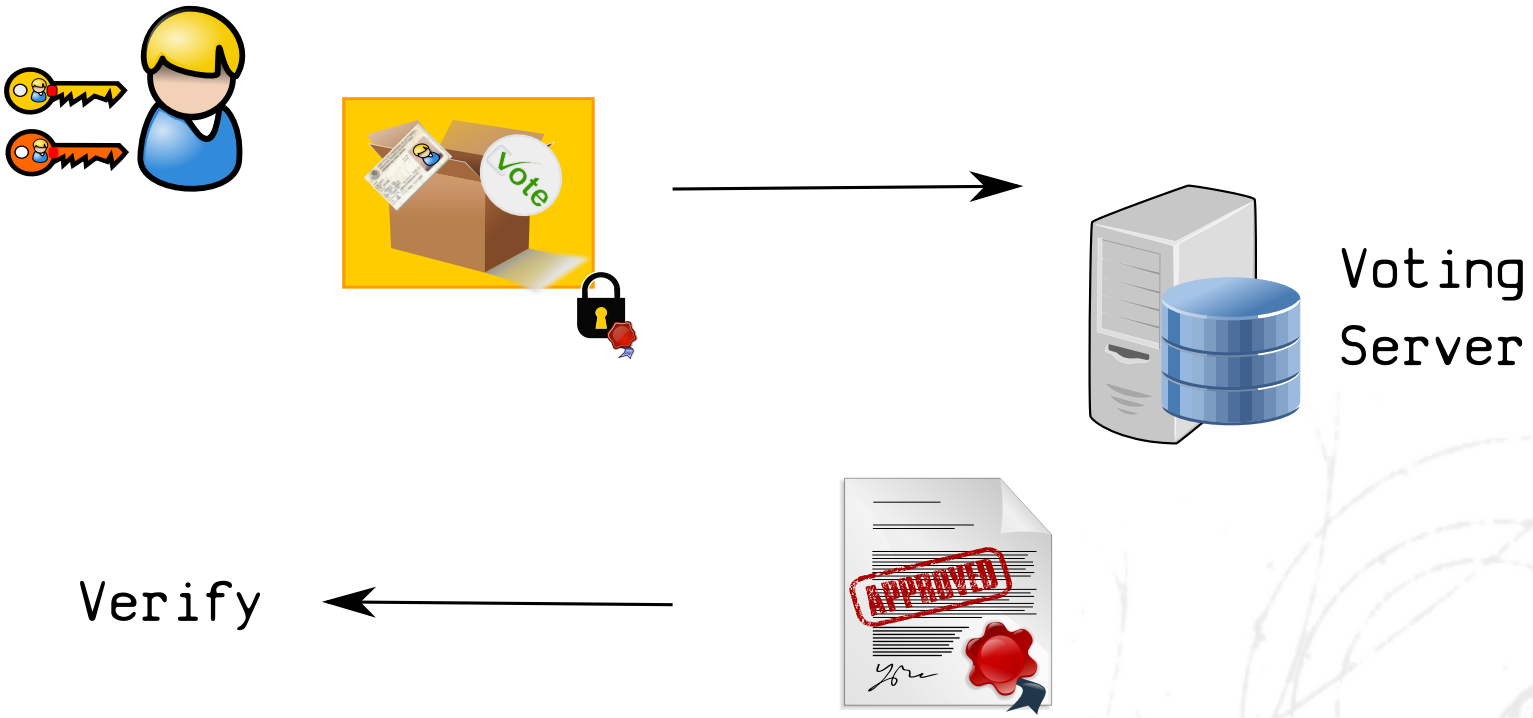
e-voting system based on pairings

An e-voting system based on short and blind signatures by Lopez-Garcia and Rodriguez-Henriquez.



e-voting system based on pairings

With a blind signature, we can cast our vote in the blank ballot.



Conclusions

- The fastest pairing function is not the panacea for some protocols
- Efficient simultaneous pairing implementation is the key to some protocols
- Scalar point multiplication is also a key primitive.
- Protocols with $\mathbb{G}_1 \neq \mathbb{G}_2$ can also be implemented efficiently
- PBC is cheaper than other solutions
- We can do 5 S.M. in $\mathbb{G}_1$, 3 in $\mathbb{G}_2$, and less than 2 expo. in $\mathbb{G}_T$ at approximately the same cost of a pairing

Future work

- More protocol/pairing implementations
- Evaluate the pairing function in a protocol context
- Evaluate the ancillary functions around the pairing
- Pairings with $\mathbb{G}_1 \neq \mathbb{G}_2$ should be encouraged
- Encourage PBC
- Consider multicore environment (ongoing work)
- More optimizations on the ancillary functions.
- Lean modular reduction (a.k.a. Lazy reduction)

Question time

- Thank you for your attention.

Contenido, sección 4

Introduction

Pairings in cryptography

Pairing protocols

Appendix

Timings

Using a Intel Core i7 2600K, Sandy Bridge

Operation	Clock cycles
RegularPairing	2108 Kclk
New Pairing	1550 Kclk
G1mul K	232.89Kclk
G1mul U	304.44Kclk
G2mul K	378.26Kclk
G2mul U	535.69Kclk
GTexpo K	617.32Kclk
GTexpo U	931.98Kclk

Detailed timings

Operation	Clock cycles
G1 Add JJA	1.92Kclk
G1 Add JJJ	2.44Kclk
G1 Dbl A	1.20Kclk
G1 Dbl J	1.44Kclk
G2 Add JJA	5.11Kclk
G2 Add JJJ	6.70Kclk
G2 Dbl A	3.03Kclk
G2 Dbl J	2.92Kclk
GT Sqr	3.78Kclk
GT Mul	9.55Kclk

Timings of Water's CPABE

LSSS ABE Protocole	CPU cycles		
	Theoretical	Expected	Measured
Encrypt	5 922 K	6 105 K	6 378 K
Keygen	1 989 K	2 014 K	2 114 K
Decrypt ($\Delta = 1$)	8 716 K	9 101 K	9 489 K
Decrypt ($\Delta \neq 1$)	9 612 K	10 051 K	10 438 K

Table : Cost of the protocol steps

Timings of the e-voting protocol

Scheme	# Cryptographic operation	# Cycles
Kharchineh & Ettelace	4 RSA-public	6,053,528
	6 RSA-private	253,251,894
	4 DLP-exponentiations	87,135,920
	Total	**346,441,342**
Li et al.	15 RSA-public	22,700,730
	9 RSA-private	379,877,841
	Total	**402,578,571**
Chung & Wu	5 RSA-public	7,566,910
	4 RSA-private	168,834,596
	Total	**176,401,506**
The proposed scheme	1 scalar multiplication in $\mathbb{G}_2$	380,000
	6 scalar multiplications in $\mathbb{G}_1$	1,800,000
	6 map-to-point functions H_1	1,890,000
	8 bilinear pairings	14,630,000
	Total	**18,700,000**

Detailed e-voting system... 1/2

An e-voting system based on short and blind signatures by Lopez-Garcia and Rodriguez-Henriquez.

Voter		Authentication Server (AS)
	Authentication phase	
$b, d_t \in \mathbb{Z}_r$		
$V_t = d_t Q \in \mathbb{G}_2$		
$m = m2s(V_t) \in \{0,1\}^{1016}$		
$\tilde{M} = bH_1(m) \in \mathbb{G}_1$		
$S_{\tilde{M}} = d_V \tilde{M} \in \mathbb{G}_1$		
	$\{ID_V, t, \tilde{M}, S_{\tilde{M}}\}$ $\longrightarrow$	
		$e(Q, S_{\tilde{M}}) \stackrel{?}{=} e(V_V, \tilde{M})$
		$\tilde{S} = d_{AS}\tilde{M} \in \mathbb{G}_1$
	$\{t, \tilde{S}\}$ $\longleftarrow$	
$S_{V_t} = b^{-1}\tilde{S} \in \mathbb{G}_1$		

... detailed e-voting system 2/2

Voting phase

		Voting server (VS)
$S_v = d_t H_1(v) \in \mathbb{G}_1$		
$B = \{V_t, S_{V_t}, v, S_v\}$	$\xrightarrow{\{B\}}$	$m = m2s(V_t)$
		$e(Q, S_{V_t}) \stackrel{?}{=} e(V_{AS}, H_1(m))$
		$e(Q, S_v) \stackrel{?}{=} e(V_t, H_1(v))$
		$a \in \mathbb{Z}_r$
		$ACK = H(V_t \|\| S_{V_t} \|\| v \|\| S_v \|\| a)$
		$S_{ACK} = d_{VS} H_1(ACK)$
	$\xleftarrow{\{ACK, S_{ACK}\}}$	
$e(Q, S_{ACK}) \stackrel{?}{=} e(V_{VS}, H_1(ACK))$		