

# Hashing into Elliptic Curves

Jean-Sébastien Coron

University of Luxembourg

# Outline

- 1 Hashing into Elliptic Curves
  - Motivation
  - Classical Techniques
- 2 Icart's Function
- 3 The SWU Algorithm
  - Background
  - SWU Algorithm
- 4 Hashing like a Random Oracle

# Introduction

- Hashing into Elliptic Curves
  - Boneh-Franklin IBE:  $Q_{id} = H_1(id)$  on the curve.
  - Password based authentication protocols (SPEKE, PAK).
- Boneh-Franklin: super-singular curve
  - Special curve with special operation: pairing.
  - Hashing is easy.
  - But larger parameters are required.
- How to hash into ordinary curves ?

# Introduction

- Hashing into Elliptic Curves
  - Boneh-Franklin IBE:  $Q_{id} = H_1(id)$  on the curve.
  - Password based authentication protocols (SPEKE, PAK).
- Boneh-Franklin: super-singular curve
  - Special curve with special operation: pairing.
  - Hashing is easy.
  - But larger parameters are required.
- How to hash into ordinary curves ?

# Introduction

- Hashing into Elliptic Curves
  - Boneh-Franklin IBE:  $Q_{id} = H_1(id)$  on the curve.
  - Password based authentication protocols (SPEKE, PAK).
- Boneh-Franklin: super-singular curve
  - Special curve with special operation: pairing.
  - Hashing is easy.
  - But larger parameters are required.
- How to hash into ordinary curves ?

# SPEKE

- Simple Password Exponential Key Exchange (Jablon, 1996)
  - Let  $pw$  be a password shared by Alice and Bob
  - Let  $E$  be the subgroup of an elliptic curve of order  $q$ .
- Protocol
  - Alice sends  $A = a.H(pw)$  to Bob, where  $a \leftarrow \mathbb{Z}_q$
  - Bob sends  $B = b.H(pw)$  to Alice, where  $b \leftarrow \mathbb{Z}_q$
  - Alice computes  $K = a.B = ab.H(pw)$
  - Bob computes  $K = b.A = ab.H(pw)$

## Try and Increment

- Elliptic curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

- Try and Increment:

Input:  $u$  an integer. We can take  $u = H(m)$ .

Output:  $Q$ , a point of  $E_{a,b}(\mathbb{F}_p)$ .

- 1 For  $i = 0$  to  $k - 1$ 
    - 1 Set  $x = u + i$
    - 2 If  $x^3 + ax + b$  is a quadratic residue in  $\mathbb{F}_p$ , then return  $Q = (x, (x^3 + ax + b)^{1/2})$
  - 2 end For
  - 3 Return  $\perp$
- Timing Attack

## Try and Increment

- Elliptic curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

- Try and Increment:

Input:  $u$  an integer. We can take  $u = H(m)$ .

Output:  $Q$ , a point of  $E_{a,b}(\mathbb{F}_p)$ .

- 1 For  $i = 0$  to  $k - 1$ 
    - 1 Set  $x = u + i$
    - 2 If  $x^3 + ax + b$  is a quadratic residue in  $\mathbb{F}_p$ , then return  $Q = (x, (x^3 + ax + b)^{1/2})$
  - 2 end For
  - 3 Return  $\perp$
- Timing Attack



# Supersingular Elliptic Curve

- Supersingular curve:

$$E : y^2 = x^3 + 1 \pmod{p}$$

- with  $p \equiv 2 \pmod{3}$
- It has  $p + 1$  points.
- Hashing into  $E$ :
  - Let  $y = H(m)$
  - Let  $x = (y^2 - 1)^{1/3}$
  - Return  $P = (x, y)$
- $p$  must be large because of MOV attack (at least 512 bits)

# Supersingular Elliptic Curve

- Supersingular curve:

$$E : y^2 = x^3 + 1 \pmod{p}$$

- with  $p \equiv 2 \pmod{3}$
- It has  $p + 1$  points.
- Hashing into  $E$ :
  - Let  $y = H(m)$
  - Let  $x = (y^2 - 1)^{1/3}$
  - Return  $P = (x, y)$
- $p$  must be large because of MOV attack (at least 512 bits)

# Supersingular Elliptic Curve

- Supersingular curve:

$$E : y^2 = x^3 + 1 \pmod{p}$$

- with  $p \equiv 2 \pmod{3}$
- It has  $p + 1$  points.
- Hashing into  $E$ :
  - Let  $y = H(m)$
  - Let  $x = (y^2 - 1)^{1/3}$
  - Return  $P = (x, y)$
- $p$  must be large because of MOV attack (at least 512 bits)

# Hashing into Ordinary Curves

- Elliptic curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

- Icart's function
  - Published by Thomas Icart at CRYPTO 2009
  - Deterministic function into  $E$
  - Requires  $p \equiv 2 \pmod{3}$
  - Essentially one exponentiation in  $\mathbb{F}_p$
- Shallue-Woestijne-Ulas algorithm
  - Deterministic algorithm into  $E$  (but requires a test)
  - Does not require  $p \equiv 2 \pmod{3}$
  - Essentially one exponentiation in  $\mathbb{F}_p$

# Hashing into Ordinary Curves

- Elliptic curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

- Icart's function

- Published by Thomas Icart at CRYPTO 2009
- Deterministic function into  $E$
- Requires  $p \equiv 2 \pmod{3}$
- Essentially one exponentiation in  $\mathbb{F}_p$

- Shallue-Woestijne-Ulas algorithm

- Deterministic algorithm into  $E$  (but requires a test)
- Does not require  $p \equiv 2 \pmod{3}$
- Essentially one exponentiation in  $\mathbb{F}_p$

# Hashing into Ordinary Curves

- Elliptic curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

- Icart's function
  - Published by Thomas Icart at CRYPTO 2009
  - Deterministic function into  $E$
  - Requires  $p \equiv 2 \pmod{3}$
  - Essentially one exponentiation in  $\mathbb{F}_p$
- Shallue-Woestijne-Ulas algorithm
  - Deterministic algorithm into  $E$  (but requires a test)
  - Does not require  $p \equiv 2 \pmod{3}$
  - Essentially one exponentiation in  $\mathbb{F}_p$

## Icart's Function

- Elliptic curve with  $p = 2 \pmod 3$ :

$$E_{a,b} : y^2 = x^3 + ax + b \pmod p$$

- Icart's function: (we can have  $u = H(m)$ )

$$\begin{aligned} f_{a,b} : \mathbb{F}_p &\mapsto E_{a,b} \\ u &\mapsto (x, y) \end{aligned}$$

$$\begin{aligned} x &= \left( v^2 - b - \frac{u^6}{27} \right)^{(2p-1)/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

$$v = \frac{3a - u^4}{6u}.$$

## Icart's Function

- Elliptic curve with  $p = 2 \pmod 3$ :

$$E_{a,b} : y^2 = x^3 + ax + b \pmod p$$

- Icart's function: (we can have  $u = H(m)$ )

$$\begin{aligned} f_{a,b} : \mathbb{F}_p &\mapsto E_{a,b} \\ u &\mapsto (x, y) \end{aligned}$$

$$\begin{aligned} x &= \left( v^2 - b - \frac{u^6}{27} \right)^{(2p-1)/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

$$v = \frac{3a - u^4}{6u}.$$



## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$



## Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let  $y = ux + v$  with  $u, v$  two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want:  $a - 2uv - u^4/3 = 0$ 
  - We take  $v = (3a - u^4)/(6u)$
- We get:  $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$x = \left( v^2 - b - \frac{u^6}{27} \right)^{1/3} + \frac{u^2}{3}$$
$$y = ux + v$$

## Quadratic residues and square-roots

### Definition

Let  $p$  be a prime.  $a \in \mathbb{Z}_p^*$  is said to be a quadratic residue modulo  $p$ , or a square modulo  $p$ , if there exists an  $x \in \mathbb{Z}_p^*$  such that  $x^2 = a \pmod{p}$ . If no such  $x$  exists, then  $a$  is called a quadratic non-residue modulo  $p$ . The set of all quadratic residues modulo  $p$  is denoted by  $Q_p$ , and the set of all quadratic non-residues is denoted  $\bar{Q}_p$

### Definition

Let  $a \in Q_p$ . If  $x \in \mathbb{Z}_p$  satisfies  $x^2 = a \pmod{p}$ , then  $x$  is called a square root of  $a$  modulo  $p$ .

## Facts

- Let  $p$  be an odd prime and let  $g$  be a generator of  $\mathbb{Z}_p^*$ .
  - $a \in \mathbb{Q}_p \Leftrightarrow a = g^{2i} \pmod p$  for some  $i \in \mathbb{Z}$ .
  - $|\mathbb{Q}_p| = (p-1)/2$  and  $|\bar{\mathbb{Q}}_p| = (p-1)/2$ .
- If  $p$  is an odd prime and  $a \in \mathbb{Q}_p$ , then  $a$  has exactly two square roots modulo  $p$ .
- Let  $p$  be an odd prime with  $p \equiv 3 \pmod 4$  and let  $a \in \mathbb{Q}_p$ . Then  $x$  and  $-x$  are the two square roots of  $a$ , where:

$$x = a^{(p+1)/4} \pmod p$$

## Legendre symbol

### Definition

The Legendre symbol with respect to an odd prime  $p$  is defined by:

$$\left(\frac{x}{p}\right) = \begin{cases} 1 & \text{if } x \not\equiv 0 \pmod{p} \text{ and } x \text{ is a square modulo } p \\ 0 & \text{if } x \equiv 0 \pmod{p} \\ -1 & \text{otherwise.} \end{cases}$$

### Fact

Let  $p \neq 2$  be a prime. For any integer  $x$ ,

$$\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \pmod{p}$$

# Shallue-Woestijne-Ulas algorithm

- Shallue-Woestijne published at ANTS 2006
  - Simplified by Ulas in 2007
  - Simplified by Icart in 2009.

## Theorem (Simplified Ulas maps)

Let  $\mathbb{F}_q$  be a field and let  $g(x) := x^3 + ax + b$ , where  $ab \neq 0$ . Let:

$$X_2(t) = \frac{-b}{a} \left( 1 + \frac{1}{t^4 - t^2} \right), \quad X_3(t) = -t^2 X_2(t), \quad U(t) = t^3 g(X_2(t))$$

Then  $U(t)^2 = -g(X_2(t)) \cdot g(X_3(t))$

- When  $q \equiv 3 \pmod{4}$ :
  - $-1$  is not a square
  - either  $g(X_2(t))$  or  $g(X_3(t))$  must be a square

# Shallue-Woestijne-Ulas algorithm

- Shallue-Woestijne published at ANTS 2006
  - Simplified by Ulas in 2007
  - Simplified by Icart in 2009.

## Theorem (Simplified Ulas maps)

Let  $\mathbb{F}_q$  be a field and let  $g(x) := x^3 + ax + b$ , where  $ab \neq 0$ . Let:

$$X_2(t) = \frac{-b}{a} \left( 1 + \frac{1}{t^4 - t^2} \right), \quad X_3(t) = -t^2 X_2(t), \quad U(t) = t^3 g(X_2(t))$$

Then  $U(t)^2 = -g(X_2(t)) \cdot g(X_3(t))$

- When  $q \equiv 3 \pmod{4}$ :
  - $-1$  is not a square
  - either  $g(X_2(t))$  or  $g(X_3(t))$  must be a square

# Simplified SWU algorithm

Simplified SWU algorithm:

Input:  $\mathbb{F}_q$  such that  $q \equiv 3 \pmod{4}$ , parameters  $a, b$  and input  $t \in \mathbb{F}_q$ . We can have  $t = H(m)$

Output:  $(x, y) \in E_{a,b}(\mathbb{F}_q)$

- 1  $\alpha \leftarrow -t^2$
- 2  $X_2 \leftarrow \frac{-b}{a} \left( 1 + \frac{1}{\alpha^2 + \alpha} \right)$
- 3  $X_3 \leftarrow \alpha \cdot X_2$
- 4  $h_2 \leftarrow (X_2)^3 + a \cdot X_2 + b$ ;  $h_3 \leftarrow (X_3)^3 + a \cdot X_3 + b$
- 5 If  $h_2$  is a square, return  $(X_2, h_2^{(q+1)/4})$ , otherwise return  $(X_3, h_3^{(q+1)/4})$

## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p = 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$



## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p \equiv 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$

## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p \equiv 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$

## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p \equiv 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$

## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p \equiv 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$

## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p \equiv 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$

## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p \equiv 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$

## Why it works

- $E : y^2 = x^3 + ax + b \pmod p$
- Let  $g(x) = x^3 + ax + b$
- Let  $u$  be a non-quadratic residue and consider the equation in  $x$ :

$$g(u \cdot x) = u^3 \cdot g(x)$$

- We can solve for  $x$ :
  - $(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) = u^3x^3 + u^3ax + u^3b$
  - $x \cdot a(u - u^3) = b(u^3 - 1) \Rightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)}$
- Since  $u$  is not a square, either  $g(u \cdot x)$  or  $g(x)$  must be a square
- When  $p \equiv 3 \pmod 4$ , we can take  $u = -t^2 \pmod p$

# Hashing like a Random Oracle

- Random Oracle Model:
  - Idealized model of computation in which the hash function is seen as a random oracle
  - Uniformly distributed output for any input
  - Many schemes proven secure in the ROM: Boneh-Franklin, etc.
- $H(m) = f_{a,b}(h(m))$  does not behave as a random oracle into the curve, even if  $h$  is a random oracle.
- Random oracle into the curve:

$$H(m) = f_{a,b}(h_1(m)) + h_2(m).G$$

- See *An Indifferentiable Hash Function into Elliptic Curves*, Jean-Sébastien Coron and Thomas Icart, <http://eprint.iacr.org/2009/340>



# Hashing like a Random Oracle

- Random Oracle Model:
  - Idealized model of computation in which the hash function is seen as a random oracle
  - Uniformly distributed output for any input
  - Many schemes proven secure in the ROM: Boneh-Franklin, etc.
- $H(m) = f_{a,b}(h(m))$  does not behave as a random oracle into the curve, even if  $h$  is a random oracle.
- Random oracle into the curve:

$$H(m) = f_{a,b}(h_1(m)) + h_2(m).G$$

- See *An Indifferentiable Hash Function into Elliptic Curves*, Jean-Sébastien Coron and Thomas Icart, <http://eprint.iacr.org/2009/340>

# Hashing like a Random Oracle

- Random Oracle Model:
  - Idealized model of computation in which the hash function is seen as a random oracle
  - Uniformly distributed output for any input
  - Many schemes proven secure in the ROM: Boneh-Franklin, etc.
- $H(m) = f_{a,b}(h(m))$  does not behave as a random oracle into the curve, even if  $h$  is a random oracle.
- Random oracle into the curve:

$$H(m) = f_{a,b}(h_1(m)) + h_2(m).G$$

- See *An Indifferentiable Hash Function into Elliptic Curves*, Jean-Sébastien Coron and Thomas Icart, <http://eprint.iacr.org/2009/340>

# Hashing like a Random Oracle

- Random Oracle Model:
  - Idealized model of computation in which the hash function is seen as a random oracle
  - Uniformly distributed output for any input
  - Many schemes proven secure in the ROM: Boneh-Franklin, etc.
- $H(m) = f_{a,b}(h(m))$  does not behave as a random oracle into the curve, even if  $h$  is a random oracle.
- Random oracle into the curve:

$$H(m) = f_{a,b}(h_1(m)) + h_2(m).G$$

- See *An Indifferentiable Hash Function into Elliptic Curves*, Jean-Sébastien Coron and Thomas Icart, <http://eprint.iacr.org/2009/340>