

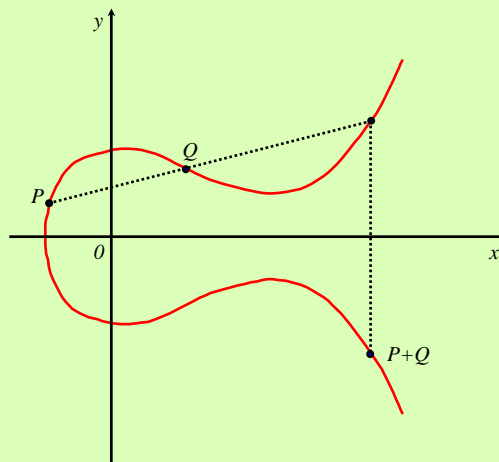
Efficient Signcryption Schemes on Elliptic Curves

Yuliang Zheng, Monash University,
Australia

Hideki Imai, University of Tokyo,
Japan

(C) 1997 by Yuliang Zheng <http://www.pscit.monash.edu.au/~yuliang/> 1

Elliptic Curve over a Finite Field $GF(p^m)$



- The points on an EC, together with the point at infinity, form an abelian group under “addition” defined by the “tangent and chord” method.
- The number of points on an elliptic curve C over $\text{GF}(p^m)$ is

$$\#C = p^m + 1 - t, \quad \text{where } |t| \leq 2\sqrt{p^m}$$

- t is called the trace of the curve

- **super-singular curves** whose traces satisfy

$$t = \pm \sqrt{i p^m},$$

where $i = 0, 1, 2, 3$ or 4

(Menezes, Okamoto & Vanstone, 93)

- **curves over $\text{GF}(p)$ with trace 1**, namely

$$\#C = p$$

(Sato & Araki, & Smart, 97)

Signcryption on EC-- public & secret parameters

- **Public to all**

- ❖ C : an EC over $GF(p^m)$,
- ❖ q : a large prime
- ❖ G : a point on C with order q
- ❖ hash, KH , (E,D)

- **Alice's keys**

- ❖ v_a : secret key
- ❖ P_a : public key
(note : $P_a = v_a G$)

- **Bob's keys**

- ❖ v_b : secret key
- ❖ P_b : public key
(note : $P_b = v_b G$)

Signcryption on EC-- 1st example

$$m \longrightarrow (c,r,s) \qquad (c,r,s) \longrightarrow m$$

- **Signcrypt by Alice**

- ❖ $k = \text{hash}(v P_b)$
where $v \in_R \{1, \dots, q-1\}$
- ❖ $k \begin{matrix} \longrightarrow & k_1 \\ & \searrow \\ & k_2 \end{matrix}$
- ❖ $r = KH_{k_2}(m)$
- ❖ $s = \frac{v}{r + v_a} \text{ mod } q$
- $c = E_{k_1}(m)$
- ❖ **output** (c,r,s)

- **Unsigncrypt by Bob**

- ❖ $u = s v_b \text{ mod } q$
 - $k = \text{hash}(u P_a + u r G)$
 - ❖ $k \begin{matrix} \longrightarrow & k_1 \\ & \searrow \\ & k_2 \end{matrix}$
 - ❖ $m = D_{k_1}(c)$
 - ❖ **output**
- $$\begin{cases} m & \text{if } r = KH_{k_2}(m) \\ \text{"invalid"} & \text{if } r \neq KH_{k_2}(m) \end{cases}$$

Signcryption on EC-- 2nd example

$$m \longrightarrow (c,r,s)$$

$$(c,r,s) \longrightarrow m$$

- **Signcrypt by Alice**

- ❖ $k = \text{hash}(v P_b)$

where $v \in_R \{1, \dots, q-1\}$

- ❖ $k \begin{matrix} \longrightarrow & k_1 \\ & \longrightarrow & k_2 \end{matrix}$

- ❖ $r = KH_{k_2}(m)$

- ❖ $s = \frac{v}{1 + v_a r} \text{ mod } q$

$$c = E_{k_1}(m)$$

- ❖ **output** (c,r,s)

- **Unsigncrypt by Bob**

- ❖ $u = s v_b \text{ mod } q$

- ❖ $k = \text{hash}(u G + u r P_a)$

- ❖ $k \begin{matrix} \longrightarrow & k_1 \\ & \longrightarrow & k_2 \end{matrix}$

- ❖ $m = D_{k_1}(c)$

- ❖ **output**

$$\begin{cases} m & \text{if } r = KH_{k_2}(m) \\ \text{"invalid"} & \text{if } r \neq KH_{k_2}(m) \end{cases}$$

EC Signcryption v.s. EC Signature-then-Encryption

- **Reduction in comp. cost**

$$\frac{5.17 - 2.17}{5.17} = 58\%$$

- **Reduction in comm. overhead**

$$\frac{|\text{hash}(\cdot)| + 2|q| - (|KH(\cdot)| + |q|)}{|\text{hash}(\cdot)| + 2|q|} = \frac{|q|}{\frac{1}{2}|q| + 2|q|} = 40\%$$