Edwards Curves and Fault Attacks

Marc Joye

technicolor

Edwards Curves and Fault Attacks

Marc Joye

technicolor

Shamir's Trick

Most countermeasures against fault attacks are variants of Shamir's trick (EUROCRYPT '97, rump session)

- Example: Shamir's trick applied to standard RSA signatures

Input: $x = h(m), d, N$
Output: $S = x^d \bmod N$

1. Choose a (small) random r
2. Compute
   1. $S^* \leftarrow x^d \bmod Nr$
   2. $Y \leftarrow x^d \bmod r$
3. If $(S^* \not\equiv Y \pmod r)$ then return `error`
4. Return $S = S^* \bmod N$

Edwards Curves

$$\mathcal{E}_{/\mathbb{F}_p} : ax^2 + y^2 = 1 + bx^2y^2 \quad \text{where } ab(a-b) \neq 0$$

- Addition law
  - $\boldsymbol{O} = (0, 1)$ [neutral element]
  - $-(x_1, y_1) = (-x_1, y_1)$
  - $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$ where

$$x_3 = \frac{x_1y_2 + x_2y_1}{1 + bx_1x_2y_1y_2}, \quad y_3 = \frac{y_1y_2 - ax_1x_2}{1 - bx_1x_2y_1y_2}$$

  - . . . also valid for point doubling (and $\boldsymbol{O}$)
- Addition law is *complete* if a is a square and b is a non-square

Shamir's Trick for Elliptic Curve Cryptosystems

$\boldsymbol{P} = (x_1, y_1) \in \mathcal{E}_{/\mathbb{F}_p} : ax^2 + y^2 = 1 + bx^2y^2$

- Let $\mathcal{R} = \mathbb{Z}/pr\mathbb{Z}$ for a (small) random prime r
  1. Compute
     -
     - $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P} \in \mathcal{E}_{pr}(\mathbb{Z}/pr\mathbb{Z})$
     - $\boldsymbol{Y} \leftarrow [d]\boldsymbol{P} \in \mathcal{E}(\mathbb{F}_r)$
  2. If ($\boldsymbol{Q}^* \not\equiv \boldsymbol{Y} \pmod r$) then return `error`
  3. Return $\boldsymbol{Q}^*$ mod p

Shamir's Trick for Elliptic Curve Cryptosystems

$\boldsymbol{P} = (x_1, y_1) \in \mathcal{E}_{/\mathbb{F}_p} : ax^2 + y^2 = 1 + bx^2y^2$

- Let $\mathcal{R} = \mathbb{Z}/pr\mathbb{Z}$ for a (small) random prime r
  1. Compute
     - $\mathcal{E}_{pr} \leftarrow \mathrm{CRT}(\mathcal{E}, \mathcal{E}_r)$ where $\mathcal{E}_{r/\mathbb{F}_r} : ax^2 + y^2 = 1 + b_r x^2 y^2$
     - $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P} \in \mathcal{E}_{pr}(\mathbb{Z}/pr\mathbb{Z})$
     - $\boldsymbol{Y} \leftarrow [d]\boldsymbol{P}_r \in \mathcal{E}_r(\mathbb{F}_r)$
  2. If ($\boldsymbol{Q}^* \not\equiv \boldsymbol{Y} \pmod r$) then return `error`
  3. Return $\boldsymbol{Q}^* \bmod p$

Idea #1

Let $b_r = (ax_1^2 + y_1^2 - 1)/(x_1^2 y_1^2) \bmod r$ so that $\boldsymbol{P}_r := \boldsymbol{P} \bmod r \in \mathcal{E}_r$

- . . . but completeness is not guaranteed (and $\#\mathcal{E}_r$ is unknown)

technicolor

Shamir's Trick for Elliptic Curve Cryptosystems

$\boldsymbol{P} = (x_1, y_1) \in \mathcal{E}_{/\mathbb{F}_p} : ax^2 + y^2 = 1 + bx^2y^2$

- Let $\mathcal{R} = \mathbb{Z}/pr\mathbb{Z}$ for a (small) ~~random~~ prime r
  1. Compute
     - $\mathcal{E}_{pr} \leftarrow \mathrm{CRT}(\mathcal{E}, \mathcal{E}_r)$ and $\boldsymbol{P}^* \leftarrow \mathrm{CRT}(\boldsymbol{P}, \boldsymbol{P}_r)$
     - $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P}^* \in \mathcal{E}_{pr}(\mathbb{Z}/pr\mathbb{Z})$
     - $\boldsymbol{Y} \leftarrow [d \pmod{n_r}]\boldsymbol{P}_r \in \mathcal{E}_r(\mathbb{F}_r)$
  2. If $(\boldsymbol{Q}^* \not\equiv \boldsymbol{Y} \pmod{r})$ then return `error`
  3. Return $\boldsymbol{Q}^* \bmod p$

Idea #2

Fix $E_r(\mathbb{F}_r) = \langle \boldsymbol{P}_r \rangle$ so that addition is complete

- . . . but r is now *a priori* fixed and values must be pre-stored

BOS$^+$ Algorithm

- Blömer, Otto, and Seifert (FDTC 2005)

Input: $\boldsymbol{P} \in \mathcal{E}, d$
Output: $\boldsymbol{Q} = [d]\boldsymbol{P}$
In memory: $\{\mathcal{E}_r, \boldsymbol{P}_r \in \mathcal{E}_r, n_r = \#\mathcal{E}_r\}$

---

1. Compute
   1. $\mathcal{E}_{pr} \leftarrow \mathrm{CRT}(\mathcal{E}, \mathcal{E}_r)$ and $\boldsymbol{P}^* \leftarrow \mathrm{CRT}(\boldsymbol{P}, \boldsymbol{P}_r)$
   2. $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P}^* \in \mathcal{E}_{pr}$ $= (x_{pr}, y_{pr})$
   3. $\boldsymbol{Y} \leftarrow [d \pmod{n_r}]\boldsymbol{P}_r \in \mathcal{E}_r$ $= (x_r, y_r)$
   4. $\begin{cases} c_x \leftarrow 1 + x_{pr} - x_r \pmod{r} \\ c_y \leftarrow 1 + y_{pr} - y_r \pmod{r} \end{cases}$
2. If $(\boldsymbol{Q}^* \not\equiv \boldsymbol{Y} \pmod{r})$ then return `error`
3. Return $\boldsymbol{Q}^* \pmod{p} \in \mathcal{E}$

BOS$^+$ Algorithm

- Blömer, Otto, and Seifert (FDTC 2005)

Input: $\boldsymbol{P} \in \mathcal{E}, d$
Output: $\boldsymbol{Q} = [d]\boldsymbol{P}$
In memory: $\{\mathcal{E}_r, \boldsymbol{P}_r \in \mathcal{E}_r, n_r = \#\mathcal{E}_r\}$

---

1. Compute
   1. $\mathcal{E}_{pr} \leftarrow \mathrm{CRT}(\mathcal{E}, \mathcal{E}_r)$ and $\boldsymbol{P}^* \leftarrow \mathrm{CRT}(\boldsymbol{P}, \boldsymbol{P}_r)$
   2. $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P}^* \in \mathcal{E}_{pr}$ $\quad = (x_{pr}, y_{pr})$
   3. $\boldsymbol{Y} \leftarrow [d \pmod{n_r}]\boldsymbol{P}_r \in \mathcal{E}_r$ $\quad = (x_r, y_r)$
   4. $\begin{cases} c_x \leftarrow 1 + x_{pr} - x_r \pmod{r} \\ c_y \leftarrow 1 + y_{pr} - y_r \pmod{r} \end{cases}$
2. For a κ-bit random ρ, compute $\gamma \leftarrow \left\lfloor \frac{\rho\, c_x + (2^\kappa - \rho) c_y)}{2^\kappa} \right\rfloor$
3. Return $\boldsymbol{Q} = [\gamma]\boldsymbol{Q}^* \pmod{p} \in \mathcal{E}$

---

technicolor

Shamir's Trick for Elliptic Curve Cryptosystems ?!

$\boldsymbol{P} = (x_1, y_1) \in \mathcal{E}_{/\mathbb{F}_p} : ax^2 + y^2 = 1 + bx^2y^2$

- Let $\mathcal{R} = \mathbb{Z}/pr\mathbb{Z}$ for a (small) random ~~prime~~ r
  1. Compute
     - $\mathcal{E}_{pr} \leftarrow \mathrm{CRT}(\mathcal{E}, \mathcal{E}_r)$ and $\boldsymbol{P}^* \leftarrow \mathrm{CRT}(\boldsymbol{P}, \boldsymbol{P}_r)$
     - $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P}^* \in \mathcal{E}_{pr}(\mathbb{Z}/pr\mathbb{Z})$
     - $\boldsymbol{Y} \leftarrow [d \pmod{n_r}]\boldsymbol{P}_r \in \mathcal{E}_r(\mathbb{Z}/r\mathbb{Z})$
  2. If $(\boldsymbol{Q}^* \not\equiv \boldsymbol{Y} \pmod{r})$ then return `error`
  3. Return $\boldsymbol{Q}^* \bmod p$

Idea #3 (???)

Choose $\mathcal{E}_r(\mathbb{Z}/r\mathbb{Z}) = \langle \boldsymbol{P}_r \rangle$, so that *(i)* addition is complete, *(ii)* $n_r = \#\mathcal{E}_r$ is known, and *(iii)* no storage is required

Proposal

$$\mathcal{E}_1(\mathbb{Z}/q^2\mathbb{Z}) = \{(\alpha q, 1) \mid \alpha \in \mathbb{Z}/q\mathbb{Z}\}$$

- Properties
  - $\mathcal{E}_1 \simeq (\mathbb{Z}/q\mathbb{Z})^+$, $\boldsymbol{P_1} = (\alpha q, 1) \overset{\sim}{\mapsto} \alpha$
  - $\#\mathcal{E}_1 = q$
  - $[d]\boldsymbol{P_1} = (dx_1, 1)$ where $x_1 = \alpha q$
- Addition law is complete

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1 y_2 + x_2 y_1}{1 + b x_1 x_2 y_1 y_2}, \frac{y_1 y_2 - a x_1 x_2}{1 - b x_1 x_2 y_1 y_2} \right)$$

whatever curve parameters a and b

Proposal

Input: $\boldsymbol{P} \in \mathcal{E}, d$
Output: $\boldsymbol{Q} = [d]\boldsymbol{P}$

1. Choose a small random t
2. Define $r \leftarrow t^2$ and $\boldsymbol{P}_r \leftarrow (t, 1)$
3. Compute
   1. $\boldsymbol{P}^* \leftarrow \mathrm{CRT}(\boldsymbol{P}, \boldsymbol{P}_r)$
   2. $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P}^* \in \mathcal{E}(\mathbb{Z}/pr\mathbb{Z})$ $= (x_{pr}, y_{pr})$
   3. $\boldsymbol{Y} \leftarrow (dt \bmod r, 1)$ $= (x_r, y_r)$
   4. $\begin{cases} c_x \leftarrow 1 + x_{pr} - x_r \pmod r \\ c_y \leftarrow y_{pr} \pmod r \end{cases}$
4. If $(\boldsymbol{Q}^* \not\equiv \boldsymbol{Y} \pmod r)$ then return `error`
5. Return $\boldsymbol{Q}^* \pmod p \in \mathcal{E}(\mathbb{F}_p)$

technicolor

Proposal

Input: $\boldsymbol{P} \in \mathcal{E}, d$
Output: $\boldsymbol{Q} = [d]\boldsymbol{P}$

1. Choose a small random t
2. Define $r \leftarrow t^2$ and $\boldsymbol{P}_r \leftarrow (t, 1)$
3. Compute
   1. $\boldsymbol{P}^* \leftarrow \mathrm{CRT}(\boldsymbol{P}, \boldsymbol{P}_r)$
   2. $\boldsymbol{Q}^* \leftarrow [d]\boldsymbol{P}^* \in \mathcal{E}(\mathbb{Z}/pr\mathbb{Z})$ $= (x_{pr}, y_{pr})$
   3. $\boldsymbol{Y} \leftarrow (dt \bmod r, 1)$ $= (x_r, y_r)$
   4. $\begin{cases} c_x \leftarrow 1 + x_{pr} - x_r \pmod r \\ c_y \leftarrow y_{pr} \pmod r \end{cases}$
4. For a κ-bit random ρ, compute $\gamma \leftarrow \left\lfloor \frac{\rho\, c_x + (2^\kappa - \rho) c_y)}{2^\kappa} \right\rfloor$
5. Return $\boldsymbol{Q} = [\gamma]\boldsymbol{Q}^* \pmod p \in \mathcal{E}(\mathbb{F}_p)$

Conclusion

- Edwards curves were introduced
  - to provide fast implementations
  - to protect against SPA-type attacks
- This talk shows that they are also useful
  - to protect against fault attacks
  - in an efficient way