Coordinate Blinding over Prime Fields

Michael Tunstall ▪ Marc Joye

Coordinate Blinding over Prime Fields

Michael Tunstall ▪ Marc Joye

technicolor

Power Leakage

The attacker aims at recovering the value of d (or a part thereof) from power traces corresponding to the computation $\boldsymbol{Q} = [d]\boldsymbol{P}$

Algorithm 1 Montgomery ladder

Input: $\boldsymbol{P} \in E(\mathbb{F}_p)$ and $d = (1, d_{\ell-2}, \ldots, d_0)_2 \in \mathbb{N}$
Output: $\boldsymbol{Q} = [d]\boldsymbol{P}$

```
R_0 ← P; R_1 ← [2]P
for j = ℓ − 2 down to 0 do
    b ← d_j; R_{1−b} ← R_{1−b} + R_b
    R_b ← [2]R_b
end for
return R_0
```

DPA-type Attack

- Let $d = (d_{\ell-1}, d_{\ell-2}, \dots, d_0)_2$
- At step j, the attacker
  - already knows bits $d_{\ell-1}, d_{\ell-2}, \dots, d_{j+1}$
  - guesses that next bit $d_j = 1$
  - chooses t random points $\boldsymbol{P_1}, \dots, \boldsymbol{P_t}$ and computes

$$\boldsymbol{X_i} = [(d_{\ell-1}, d_{\ell-2}, \dots, d_{j+1}, d_j)_2]\boldsymbol{P_i} \quad \text{for } 1 \leqslant i \leqslant t$$

  - prepares two sets

$$\mathscr{S}_0 = \{\boldsymbol{P_i} \mid g(\boldsymbol{X_i}) = 0\} \text{ and } \mathscr{S}_1 = \{\boldsymbol{P_i} \mid g(\boldsymbol{X_i}) = 1\}$$

  - $$\boxed{\text{if } \langle \mathscr{C}(i) \rangle_{\substack{1 \leqslant i \leqslant t \\ \boldsymbol{P_i} \in \mathscr{S}_0}} - \langle \mathscr{C}(i) \rangle_{\substack{1 \leqslant i \leqslant t \\ \boldsymbol{P_i} \in \mathscr{S}_1}} \begin{cases} \approx 0 & \text{then } d_j = 0 \\ \not\approx 0 & \text{then } d_j = 1 \end{cases}}$$

- Iterate the attack to find $d_{j-1}, \dots$

technicolor

Protecting Against DPA

- Known DPA-type attacks require that
  1. the crypto-device computes $\boldsymbol{Q} = [d]\boldsymbol{P}$ for a fixed d
  2. the attacker is able to evaluate

$$g(\boldsymbol{X_i}) \quad \text{with } \boldsymbol{X_i} = [(d_{\ell-1}, d_{\ell-2}, \ldots, d_{j+1}, d_j)_2]\boldsymbol{P_i}$$

  - randomization techniques aim at disabling the attacker to evaluate $g(\boldsymbol{X_i})$

This Talk

Goal

Study of randomization techniques as a countermeasure against DPA-type attacks

- coordinate blinding
- prime fields

technicolor

Outline

1. **Randomization Techniques**
   - Scalar randomization
   - Point randomization
2. **Coordinate Blinding**
   - Principle
   - Jacobian coordinates
   - Homogeneous coordinates
3. **Implementation**
   - Selecting the parameters
   - Montgomery multiplication
4. **Conclusion**

Outline

technicolor

Randomizing Scalar d

- Blinding method
  - $\#E = hn$ and $\mathrm{ord}_E(\boldsymbol{P}) = n$
  - $d^* = d + rn$

$$\boldsymbol{Q} = [d^*]\boldsymbol{P}$$

- Recoding method
  - signed-digit representations are not unique
  - for a random representation of d, say d^*, $\boldsymbol{Q} = [d^*]\boldsymbol{P}$

- Splitting methods
  1. additive: $\boldsymbol{Q} = [d - r]\boldsymbol{P} + [r]\boldsymbol{P}$
  2. multiplicative: $\boldsymbol{Q} = [d r^{-1} \pmod n]([r]\boldsymbol{P})$
  3. Euclidean: $d = \lfloor d/r \rfloor r + (d \bmod r)$

$$\Longrightarrow \boldsymbol{Q} = [d \bmod r]\boldsymbol{P} + \big[\lfloor d/r \rfloor\big]\big([r]\boldsymbol{P}\big)$$

technicolor

Randomizing Point $\boldsymbol{P}$

- Point blinding
  - let $\boldsymbol{S} = [d]\boldsymbol{R}$ for a secret point $\boldsymbol{R}$

$$\boldsymbol{Q} = [d](\boldsymbol{R} + \boldsymbol{P}) - \boldsymbol{S}$$

  - pair $(\boldsymbol{R}, \boldsymbol{S})$ in EEPROM is updated by $([t]\boldsymbol{R}, [t]\boldsymbol{S})$ for a (small) random t
- Randomized initial point
  - in the right-to-left scalar multiplication algorithms the accumulator, $\boldsymbol{R_0}$ is initialized with $\boldsymbol{O}$
  - if the initial point is a random point, say $\boldsymbol{T}$, intermediate information will be randomized
    - (of course, at the end of the computation, $\boldsymbol{T}$ should be subtracted to get the correct result)
- Drawbacks
  - storage of $(\boldsymbol{R}, \boldsymbol{S})$
  - generation of $\boldsymbol{T}$ (or its storage)

technicolor

Randomized Projective Coordinates

- Let $\boldsymbol{P} = (x_1, y_1)$
- Homogeneous coordinates

$$\boldsymbol{Q} = [d](\theta x_1 : \theta y_1 : \theta) \quad \text{for a random } \theta \neq 0$$

- Jacobian coordinates

$$\boldsymbol{Q} = [d](\theta^2 x_1 : \theta^3 y_1 : \theta) \quad \text{for a random } \theta \neq 0$$

technicolor

Randomized Curve Isomorphisms

- Let $E: y^2 = x^3 + ax + b$ and $E': y^2 = x^3 + a'x + b$

$$\begin{tikzcd} \boldsymbol{P} \in E \arrow[r, "\text{mult. by } d"] \arrow[d, "\varphi"'] & \boldsymbol{Q} = [d]\boldsymbol{P} \in E \\ \boldsymbol{P'} \in E' \arrow[r, "\text{mult. by } d"] & \boldsymbol{Q'} = [d]\boldsymbol{P'} \in E' \arrow[u, "\varphi^{-1}"'] \end{tikzcd}$$

$$\Longrightarrow \boldsymbol{Q} = \varphi^{-1}\big([d]\boldsymbol{P'}\big) \text{ with } \boldsymbol{Q'} = \varphi(\boldsymbol{P})$$

- Isomorphisms given by

$$\begin{cases} \varphi : E \to E', \boldsymbol{P} = (x_1, y_1) \mapsto \boldsymbol{P'} = (u^{-2}x_1, u^{-3}y_1) \\ \varphi^{-1} : E' \to E, \boldsymbol{P'} = (x_1', y_1') \mapsto \boldsymbol{P} = (u^2x_1', u^3y_1') \end{cases}$$

and $a' = u^{-4}a$ and $b' = u^{-6}b$ for a random $u \neq 0$

technicolor

Randomizing Point P: Comparison

Randomized curve isomorphisms technique (RCI)

- is mostly dedicated to left-to-right point multiplication algorithms
  - it allows the use of mixed point addition (i.e., $Z_2 = 1$)
- makes parameter a random (large)
  - the fastest doubling formula with $a = -3$ cannot be used

Randomized projective coordinates technique (RPC)

- is mainly useful for right-to-left point multiplication algorithms
- does modify the value of a

Can we generalize the approach?

technicolor

Outline

1. Randomization Techniques
   - Scalar randomization
   - Point randomization
2. **Coordinate Blinding**
   - Principle
   - Jacobian coordinates
   - Homogeneous coordinates
3. Implementation
   - Selecting the parameters
   - Montgomery multiplication
4. Conclusion

Coordinate Blinding

Principle

- Define the map Φ as mapping a point $\boldsymbol{P} = (X, Y, Z) \in E$ to the coordinate $\boldsymbol{P'} = \Phi(\boldsymbol{P})$ where

$$\Phi(\boldsymbol{P}) = (X', Y', Z) = (f^{\mu} X, f^{\nu} Y, Z)$$

  for an arbitrary $f \in \mathbb{F}_p \setminus \{0\}$ and some small integers μ and ν
- Algorithms for addition and doubling operations are then redefined such that $\boldsymbol{R'} = \boldsymbol{P'} + \boldsymbol{Q'}$

1. $\boldsymbol{P'}$ is not necessarily in E
2. inverse of Φ can be computed without inverting f since

Jacobian: $\boldsymbol{P} = \Phi^{-1}(\boldsymbol{P'}) = (f^{\mu+2\nu} X', f^{3\mu+2\nu} Y', f^{\mu+\nu} Z)$

Homogeneous: $\boldsymbol{P} = \Phi^{-1}(\boldsymbol{P'}) = (f^{\nu} X', f^{\mu} Y', f^{\mu+\nu} Z)$

technicolor

Jacobian Coordinates: Addition Algorithm

- For $\boldsymbol{R} = \boldsymbol{P} + \boldsymbol{Q}$, redefine the addition algorithm such that $\Phi(\boldsymbol{R}) = \boldsymbol{R}' = \boldsymbol{P}' + \boldsymbol{Q}' = \Phi(\boldsymbol{P}) + \Phi(\boldsymbol{Q})$
- Let the coordinates $\boldsymbol{R}' = (X'_3, Y'_3, Z_3)$, $\boldsymbol{P}' = (X'_1, Y'_1, Z_1)$ and $\boldsymbol{Q}' = (X'_2, Y'_2, Z_2)$
- If $\boldsymbol{P} \neq \boldsymbol{Q}$:

$$\lambda_1 = {Z_1}^2, \quad \lambda_2 = {Z_2}^2, \quad \lambda_3 = X'_1\lambda_2, \quad \lambda_4 = X'_2\lambda_1,$$
$$\lambda_5 = Y'_1 Z_2 \lambda_2, \quad \lambda_6 = Y'_2 Z_1 \lambda_1, \quad \lambda_7 = \lambda_4 - \lambda_3, \quad \lambda_8 = (2\lambda_7)^2,$$
$$\lambda_9 = \lambda_7\lambda_8, \quad \lambda_{10} = 2(\lambda_6 - \lambda_5), \quad \lambda_{11} = \lambda_3\lambda_8,$$
$$X'_3 = f^{3\mu-2\nu}{\lambda_{10}}^2 - \lambda_9 - 2\lambda_{11} \quad Y'_3 = \lambda_{10}(\lambda_{11} - X'_3) - 2\lambda_5\lambda_9$$
$$Z_3 = ((Z_1 + Z_2)^2 - \lambda_1 - \lambda_2)\lambda_7$$

  - This requires an extra multiplication in $\mathbb{F}_p$
  - Note this requires $3\mu \geqslant 2\nu$

technicolor

Jacobian Coordinates: Doubling Algorithm

- If $\boldsymbol{P} = \boldsymbol{Q}$:

$$\lambda_1 = {X'_1}^2, \quad \lambda_2 = {Y'_1}^2, \quad \lambda_3 = {\lambda_2}^2, \quad \lambda_4 = {Z_1}^2,$$
$$\lambda_5 = 2((X'_1 + \lambda_2)^2 - \lambda_1 - \lambda_3), \quad \lambda_6 = 3\lambda_1 + af^{2\mu}{\lambda_4}^2,$$
$$\lambda_7 = f^{2\nu - 3\mu}{\lambda_6}^2 - 2\lambda_5, \quad X'_3 = \lambda_7, \quad Y'_3 = f^{2\nu-3\mu}\lambda_6(\lambda_5 - \lambda_7) - 8\lambda_3,$$
$$Z_3 = (Y'_1 + Z_1)^2 - \lambda_2 - \lambda_4$$

  - This requires an extra two multiplications in $\mathbb{F}_p$
  - Note this requires $3\mu \leqslant 2\nu$

Jacobian Coordinates: Doubling Algorithm ($a = -3$)

- If $\boldsymbol{P} = \boldsymbol{Q}$ and $a = -3$:

$$\lambda_1 = {Z_1}^2, \quad \lambda_2 = {Y_1'}^2, \quad \lambda_3 = X_1'\lambda_2, \quad \lambda_4 = f^{\mu}\lambda_1,$$
$$\lambda_5 = 3(X_1' - \lambda_4)(X_1' + \lambda_4), \quad X_3' = f^{2\nu-3\mu}{\lambda_5}^2 - 8\lambda_3,$$
$$Y_3' = f^{2\nu-3\mu}\lambda_5(4\lambda_3 - X_3') - 8{\lambda_2}^2, \quad Z_3 = (Y_1' + Z_1)^2 - \lambda_2 - \lambda_1$$

  - This requires an extra three multiplications in $\mathbb{F}_p$
  - Note this requires $3\mu \leqslant 2\nu$

Homogeneous Coordinates: Addition Algorithm

- Likewise with homogeneous coordinates
- If $\boldsymbol{P} \neq \boldsymbol{Q}$:

$$\lambda_1 = Y_1' Z_2, \quad \lambda_2 = X_1' Z_2, \quad \lambda_3 = Z_1 Z_2, \quad \lambda_4 = Y_2' Z_1 - \lambda_1,$$
$$\lambda_5 = {\lambda_4}^2, \quad \lambda_6 = X_2' Z_1 - \lambda_2, \quad \lambda_7 = {\lambda_6}^2, \quad \lambda_8 = \lambda_6 \lambda_7,$$
$$\lambda_9 = \lambda_7 \lambda_2, \quad \lambda_{10} = f^{3\mu - 2\nu} \lambda_5 \lambda_3 - \lambda_8 - 2\lambda_9,$$
$$X_3' = \lambda_6 \lambda_{10}, \quad Y_3' = \lambda_4(\lambda_9 - \lambda_{10}) - \lambda_8 \lambda_1, \quad Z_3 = \lambda_8 \lambda_3$$

  - This requires an extra multiplication in $\mathbb{F}_p$
  - Note this requires $3\,\mu \geqslant 2\,\nu$

technicolor

Homogeneous Coordinates: Doubling Algorithm

- If $P = Q$:

$$\lambda_1 = {X_1'}^2, \quad \lambda_2 = {Z_1}^2, \quad \lambda_3 = a f^{2\mu} \lambda_2 + 3\lambda_1, \quad \lambda_4 = 2Y_1'Z_1,$$
$$\lambda_5 = {\lambda_4}^2, \quad \lambda_6 = \lambda_4\lambda_5, \quad \lambda_7 = Y_1'\lambda_4, \quad \lambda_8 = {\lambda_7}^2,$$
$$\lambda_9 = (X_1' + \lambda_7)^2 - \lambda_1 - \lambda_8, \lambda_{10} = f^{2\nu-3\mu} {\lambda_3}^2 - 2\lambda_9 X_3' = \lambda_{10}\lambda_4,$$
$$Y_3' = f^{2\nu-3\mu} \lambda_3(\lambda_9 - \lambda_{10}) - 2\lambda_8, \quad Z_3 = \lambda_6$$

- If $P = Q$ and $a = -3$:

$$\lambda_0 = f^{\mu} Z_1, \quad \lambda_1 = 3(X_1' - \lambda_0)(X_1' + \lambda_0), \quad \lambda_2 = 2Y_1'Z_1,$$
$$\lambda_3 = {\lambda_2}^2, \quad \lambda_4 = \lambda_2\lambda_3, \quad \lambda_5 = Y_1'\lambda_2, \quad \lambda_6 = {\lambda_5}^2,$$
$$\lambda_7 = 2X_1'\lambda_5, \quad \lambda_8 = f^{2\nu-3\mu} {\lambda_1}^2 - 2\lambda_7, \quad X_3' = \lambda_8\lambda_2$$
$$Y_3' = f^{2\nu-3\mu} \lambda_1(\lambda_7 - \lambda_8) - 2\lambda_6, \quad Z_3 = \lambda_4$$

  - This requires an extra two (three) multiplications in $\mathbb{F}_p$
  - Note this requires $3\mu \leqslant 2\nu$

technicolor

Outline

1 Randomization Techniques
- Scalar randomization
- Point randomization

2 Coordinate Blinding
- Principle
- Jacobian coordinates
- Homogeneous coordinates

3 **Implementation**
- **Selecting the parameters**
- **Montgomery multiplication**

4 Conclusion

Choosing Parameters μ and ν

- Given the constraints a good choice for μ and ν would satisfy $3\mu = 2\nu$
  - addition algorithms then requires no extra multiplication
  - doubling algorithms require one multiplication with
    - $af^{2\mu}$,
    - or one extra with f^{μ} if $a = -3$

  (for both Jacobian and homogeneous coordinates)

- For $(\mu, \nu) = (2, 3)$, resulting algorithms are equivalent to choosing a curve isomorphism given by

$$\psi : E \xrightarrow{\sim} E^* : \begin{cases} \boldsymbol{P} = (x, y) \mapsto \boldsymbol{P^*} = (f^2 x, f^3 y) \\ \boldsymbol{O} \mapsto \boldsymbol{O} \end{cases}$$

technicolor

Montgomery Multiplication

- Implementing these operations will typically make use of Montgomery multiplication

Algorithm 2 Montgomery multiplication

Input: $m = (m_{n-1}, \ldots, m_1, m_0)_b$, $x = (x_{n-1}, \ldots, x_1, x_0)_b$, $y = (y_{n-1}, \ldots, y_1, y_0)_b$ with $0 \leqslant x, y < m$, $R = b^n$, $\gcd(m, b) = 1$ and $m' = -m^{-1} \bmod b$

Output: $A = x\,y\,R^{-1} \bmod m$

```
A ← 0
for i = 0 to n − 1 do
    u_i ← (a_0 + x_i y_0) m' mod b
    A ← (A + x_i y + u_i m) / b
end for
if A ⩾ m then A ← A − m
return A
```

  - requires $n(2n+2)$ single-precision multiplications

Montgomery Multiplication with a Word

- Assume $x = x_0 < b$

Algorithm 3 Montgomery mult. with a word

Input: $m = (m_{n-1}, \ldots, m_1, m_0)_b$, $x_0 \in \{0, \ldots, b-1\}$, $y = (y_{n-1}, \ldots, y_1, y_0)_b$ with $0 \leqslant y < m$, $\gcd(m, b) = 1$ and $m' = -m^{-1} \bmod b$

Output: $A = x_0 y b^{-1} \bmod m$

1: $u \leftarrow x_0 y_0 m' \bmod b$
2: $A \leftarrow (x_0 y + u m)/b$
3: **if** $A \geqslant m$ **then** $A \leftarrow A - m$
4: **return** A

- requires $2n + 2$ single-precision multiplications
- result is multiplied by $b^{-1} \bmod m$

technicolor

Montgomery Multiplication with f

- Define the random value that is used to be

$$f' \equiv bf \pmod{m}$$

with $f' \in \{1,\ldots,b-1\}$ and $f \in \mathbb{F}_p \setminus \{0\}$

  - in practice this means f' is a random value in $\{1,\ldots,b-1\}$
  - multiplying with f costs $2n+2$ single multiplications and multiplying with f^μ costs $(2n+2)\mu$ single multiplications

- (This could equally be used to efficiently randomize the representation of a projective coordinate)

technicolor

Outline

1 **Randomization Techniques**
- Scalar randomization
- Point randomization

2 **Coordinate Blinding**
- Principle
- Jacobian coordinates
- Homogeneous coordinates

3 **Implementation**
- Selecting the parameters
- Montgomery multiplication

4 **Conclusion**

Summary

- Conceptually simple blinding method for implementing scalar multiplication over Weierstraß curves
  - shown to be equivalent to the curve isomorphisms for some parameters
- Details on how to efficiently implement this countermeasure
  - Montgomery multiplication
  - (can be applied to other existing countermeasures)

technicolor

Questions?

technicolor