Unified Point Addition Formulæ and Side-Channel Attacks

Douglas Stebila[1] and Nicolas Thériault

D.S.: Institute for Quantum Computing,
University of Waterloo, Waterloo, Ontario, Canada
N.T.: Fields Institute, Toronto, Ontario, Canada

CHES 2006 – Fri. Oct. 13, 2006

[1] Supported by Canada's NSERC, Sun Microsystems, CIAR, MITACS, CFI, and ORDCF.

Outline

Elliptic curves

- In this presentation we will work with elliptic curves in Weierstraß form over prime fields $\mathbb{F}_p$ of characteristic $p > 3$, so the curve has the form:

$$y^2 = x^3 + ax + b \mod p$$

Elliptic curves

- In this presentation we will work with elliptic curves in Weierstraß form over prime fields $\mathbb{F}_p$ of characteristic $p > 3$, so the curve has the form:

$$y^2 = x^3 + ax + b \mod p$$

- The point addition and doubling formulæ are:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \mod p, \\ y_3 &= \lambda(x_1 - x_3) - y_1 \mod p, \\ \lambda &= \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} \mod p, & \text{if } x_1 \neq x_2 \text{ (addition)}, \\ \frac{3x_1^2 + a}{2y_1} \mod p, & \text{if } x_1 = x_2 \text{ (doubling)}. \end{cases} \end{aligned}$$

Simple side-channel attacks on ECC point multiplication

Algorithm: Double-and-add point multiplication
Input: Point P, scalar $k = \sum_{j=0}^{\ell-1} k_j 2^j$.
Output: Point kP.

1. $Q \leftarrow P$.
2. If $k_{\ell-1} = 1$ then $R \leftarrow P$ else $R \leftarrow \mathcal{O}$.
3. For i from $\ell - 2$ to 0 do:
   - 3.1 **[Double]** $Q \leftarrow 2Q$.
   - 3.2 **[Add]** If $k_i = 1$ then $R \leftarrow R + Q$
4. Return R.

Simple side-channel attacks on ECC point multiplication

Algorithm: Double-and-add point multiplication
Input: Point P, scalar $k = \sum_{j=0}^{\ell-1} k_j 2^j$.
Output: Point kP.

1. $Q \leftarrow P$.
2. If $k_{\ell-1} = 1$ then $R \leftarrow P$ else $R \leftarrow \mathcal{O}$.
3. For i from $\ell - 2$ to 0 do:
   - 3.1 **[Double]** $Q \leftarrow 2Q$.
   - 3.2 **[Add]** If $k_i = 1$ then $R \leftarrow R + Q$
4. Return R.

Simple side-channel attacks on ECC point multiplication

Algorithm: Double-and-add point multiplication
Input: Point P, scalar $k = \sum_{j=0}^{\ell-1} k_j 2^j$.
Output: Point kP.

1. $Q \leftarrow P$.
2. If $k_{\ell-1} = 1$ then $R \leftarrow P$ else $R \leftarrow \mathcal{O}$.
3. For i from $\ell - 2$ to 0 do:
   3.1 **[Double]** $Q \leftarrow 2Q$.
   3.2 **[Add]** If $k_i = 1$ then $R \leftarrow R + Q$
4. Return R.

- If the double-and-add algorithm is used for point multiplication with the textbook formulæ, then it can be easy to read off the key bits if a side-channel exists that distinguishes point additions from point doublings.

Model for side-channel analysis

- Find operations in the formulæ that distinguish point addition from point doubling.

Model for side-channel analysis

- Find operations in the formulæ that distinguish point addition from point doubling.
- Use side-channel analysis to observe these distinguishing operations and identify corresponding point additions and point doublings.

Model for side-channel analysis

- Find operations in the formulæ that distinguish point addition from point doubling.
- Use side-channel analysis to observe these distinguishing operations and identify corresponding point additions and point doublings.
- Model the sequence using a Markov process and apply statistical analysis to limit the possible key space.

Model for side-channel analysis

- Find operations in the formulæ that distinguish point addition from point doubling.
- Use side-channel analysis to observe these distinguishing operations and identify corresponding point additions and point doublings.
- Model the sequence using a Markov process and apply statistical analysis to limit the possible key space.
- Perform a brute-force search on the remaining key space.

Defence techniques

- Defence against simple side-channel analysis:
  - insert dummy operations,
  - use multiplication algorithms that behave regularly, or
  - unify point addition and doubling formulæ or consider other parameterizations.

Results

- We give a projective version of the unified point addition formulæ of [BDJ04].

Results

- We give a projective version of the unified point addition formulæ of [BDJ04].
- We apply an extension of an attack of [Wal04] to these affine and projective formulæ and show that it is a feasible attack.

Results

- We give a projective version of the unified point addition formulæ of [BDJ04].
- We apply an extension of an attack of [Wal04] to these affine and projective formulæ and show that it is a feasible attack.
- We suggest countermeasures to avoid these attacks.

Goal of the attack

- A large number of keys are used for scalar multiplication (each key is used only once). We want to find some (but not all) of the keys used.
- The attack is successful if a non-negligible proportion of the keys can be computed at a cost which is considerably lower than before the attack.

Goal of the attack

- A large number of keys are used for scalar multiplication (each key is used only once). We want to find some (but not all) of the keys used.
- The attack is successful if a non-negligible proportion of the keys can be computed at a cost which is considerably lower than before the attack.
- Walter [Wal04] looks for the most easily computable key out of an (average) set of 512 keys.
- This corresponds to observing 512 traces of point multiplication using different keys and then choosing the trace that is most susceptible to attack.

Assumptions of attack

- Point multiplication implemented using double-and-add.
- Field multiplication is done using Montgomery modular multiplication with conditional subtraction.
- A conditional subtraction can be detected.

Assumptions of attack

- Point multiplication implemented using double-and-add.
- Field multiplication is done using Montgomery modular multiplication with conditional subtraction.
- A conditional subtraction can be detected.
- These assumptions are justifiable:
  - Double-and-add style multiplication is often used in memory-constrained environments.
  - Montgomery modular multiplication with conditional subtraction is widely used.

Montgomery modular multiplication

Algorithm: Montgomery modular multiplication
Input: A, B, P such that $A, B < R \leq r^{-n}$, P coprime to R.
Output: C such that $C \equiv ABr^{-n} \mod P$, $C < R$.

1. $C \leftarrow AB$.
2. $C \leftarrow (C + (-p^{-1}C \mod R)p)/R$.
3. If $C \geq R$ then $C \leftarrow C - P$.

Montgomery modular multiplication

Algorithm: Montgomery modular multiplication
Input: A, B, P such that $A, B < R \leq r^{-n}$, P coprime to R.
Output: C such that $C \equiv ABr^{-n} \mod P$, $C < R$.

1. $C \leftarrow AB$.
2. $C \leftarrow (C + (-p^{-1}C \mod R)p)/R$.
3. If $C \geq R$ then $C \leftarrow C - P$.

Brier, Déchène, and Joye's unified formula – affine form

- [BDJ04] give an infinite family of unified point addition formulæ; we choose the most efficient one.

Brier, Déchène, and Joye's unified formula – affine form

- [BDJ04] give an infinite family of unified point addition formulæ; we choose the most efficient one.
- Unified form of λ:

$$\lambda = \frac{(x_1 + x_2)^2 - x_1 x_2 + a + (-1)^\delta (y_1 - y_2)}{y_1 + y_2 + (-1)^\delta (x_1 - x_2)} ,$$

where $\delta = 0$ when $y_1 + y_2 + x_1 - x_2 \neq 0$ and $\delta = 1$ otherwise.

Brier, Déchène, and Joye's unified formula – affine form

- [BDJ04] give an infinite family of unified point addition formulæ; we choose the most efficient one.
- Unified form of λ:

$$\lambda = \frac{(x_1 + x_2)^2 - x_1x_2 + a + (-1)^\delta (y_1 - y_2)}{y_1 + y_2 + (-1)^\delta (x_1 - x_2)} \quad ,$$

  where $\delta = 0$ when $y_1 + y_2 + x_1 - x_2 \neq 0$ and $\delta = 1$ otherwise.
- The form of x_3 and y_3 is not changed, only the form of λ is changed.
- This unified formula requires $y_1 + y_2 + (-1)^\delta (x_1 - x_2) \neq 0$.

Application of [Wal04]'s attack

$$\lambda = \frac{(x_1 + x_2)^2 - x_1 x_2 + a + (-1)^\delta \, (y_1 - y_2)}{y_1 + y_2 + (-1)^\delta \, (x_1 - x_2)}$$

Application of [Wal04]'s attack

$$\lambda = \frac{(x_1 + x_2)^2 - x_1 x_2 + a + (-1)^{\delta} (y_1 - y_2)}{y_1 + y_2 + (-1)^{\delta} (x_1 - x_2)}$$

- Parts of [Wal04]'s attack can be applied here.
- A modular subtraction $a - b \mod p$ is implemented as follows:
  1. $c \leftarrow a - b$.
  2. if $c < 0$ then $c \leftarrow c + p$

Application of [Wal04]'s attack

$$\lambda = \frac{(x_1 + x_2)^2 - x_1 x_2 + a + (-1)^\delta (y_1 - y_2)}{y_1 + y_2 + (-1)^\delta (x_1 - x_2)}$$

- Parts of [Wal04]'s attack can be applied here.
- A modular subtraction $a - b \mod p$ is implemented as follows:
  1. $c \leftarrow a - b$.
  2. if $c < 0$ then $c \leftarrow c + p$

Application of [Wal04]'s attack

$$\lambda = \frac{(x_1 + x_2)^2 - x_1 x_2 + a + (-1)^\delta \boxed{(y_1 - y_2)}}{y_1 + y_2 + (-1)^\delta \boxed{(x_1 - x_2)}}$$

- Parts of [Wal04]'s attack can be applied here.
- A modular subtraction $a - b \mod p$ is implemented as follows:
  1. $c \leftarrow a - b$.
  2. if $c < 0$ then $\boxed{c \leftarrow c + p}$
- If a conditional addition occurs in the computation of either $(y_1 - y_2)$ or $(x_1 - x_2)$, then the operation cannot be a point doubling.

Attack analysis assumptions

- Assume that the bit length of the key is known and is maximal (this is true roughly $1/2$ the time).
- Assume that the total number of point operations (additions and doublings) is known.

Attack analysis assumptions

- Assume that the bit length of the key is known and is maximal (this is true roughly $1/2$ the time).
- Assume that the total number of point operations (additions and doublings) is known.
- This implies we know the number of point additions that need to be identified.

Attack analysis assumptions

- Assume that the bit length of the key is known and is maximal (this is true roughly $1/2$ the time).
- Assume that the total number of point operations (additions and doublings) is known.
- This implies we know the number of point additions that need to be identified.
- Example:
  - 192-bit key with 285 point operations and 84 identified point additions.

Attack analysis assumptions

- Assume that the bit length of the key is known and is maximal (this is true roughly $1/2$ the time).
- Assume that the total number of point operations (additions and doublings) is known.
- This implies we know the number of point additions that need to be identified.
- Example:
  - 192-bit key with 285 point operations and 84 identified point additions.
  - This leaves $285 - 192 - 84 = 9$ unidentified point additions.

Attack analysis assumptions

- Assume that the bit length of the key is known and is maximal (this is true roughly $1/2$ the time).
- Assume that the total number of point operations (additions and doublings) is known.
- This implies we know the number of point additions that need to be identified.
- Example:
  - 192-bit key with 285 point operations and 84 identified point additions.
  - This leaves $285 - 192 - 84 = 9$ unidentified point additions.
  - The size of the keyspace is upper bounded by $\binom{192}{9}$.

Projective form of [BDJ04] formula

- Let $x_i = X_i/Z_i, y_i = Y_i/Z_i$. Then

$$X_3 = 2FW \quad Y_3 = R(G - 2W) - LFM \quad Z_3 = 2F^3$$

$$U_1 = X_1 Z_2 \qquad U_2 = X_2 Z_1$$

$$S_1 = Y_1 Z_2 \qquad S_2 = Y_2 Z_1$$

$$Z = Z_1 Z_2 \quad T = U_1 + U_2 \quad M = S_1 + S_2$$

$$V = (-1)^\delta (U_1 - U_2) \qquad N = (-1)^\delta (S_1 - S_2)$$

$$E = M + V \quad F = ZE \quad L = FE$$

$$R = T^2 - U_1 U_2 + Z(aZ + N) \quad W = R^2 - G$$

Projective form of [BDJ04] formula

- Let $x_i = X_i/Z_i, y_i = Y_i/Z_i$. Then

$$X_3 = 2FW \quad Y_3 = R(G - 2W) - LFM \quad Z_3 = 2F^3$$

$$U_1 = X_1Z_2 \quad U_2 = X_2Z_1$$

$$S_1 = Y_1Z_2 \quad S_2 = Y_2Z_1$$

$$Z = Z_1Z_2 \quad T = U_1 + U_2 \quad M = S_1 + S_2$$

$$V = (-1)^\delta(U_1 - U_2) \qquad N = (-1)^\delta(S_1 - S_2)$$

$$E = M + V \quad F = ZE \quad L = FE$$

$$R = T^2 - U_1U_2 + Z(aZ + N) \quad W = R^2 - G$$

Projective form of [BDJ04] formula

- Let $x_i = X_i/Z_i, y_i = Y_i/Z_i$. Then

$$X_3 = 2FW \quad Y_3 = R(G - 2W) - LFM \quad Z_3 = 2F^3$$

$$U_1 = X_1 Z_2 \quad U_2 = X_2 Z_1$$

$$S_1 = Y_1 Z_2 \quad S_2 = Y_2 Z_1$$

$$Z = Z_1 Z_2 \quad T = U_1 + U_2 \quad M = S_1 + S_2$$

$$V = (-1)^\delta (U_1 - U_2) \qquad N = (-1)^\delta (S_1 - S_2)$$

$$E = M + V \quad F = ZE \quad L = FE$$

$$R = T^2 - U_1 U_2 + Z(aZ + N) \quad W = R^2 - G$$

- If a conditional subtraction occurs in U_1 but not U_2 or vice versa, then it is not a point doubling; similarly for S_1, S_2.

Projective form of [BDJ04] formula

- Let $x_i = X_i/Z_i, y_i = Y_i/Z_i$. Then

$$X_3 = 2FW \quad Y_3 = R(G - 2W) - LFM \quad Z_3 = 2F^3$$

$$U_1 = X_1 Z_2 \qquad U_2 = X_2 Z_1$$

$$S_1 = Y_1 Z_2 \qquad S_2 = Y_2 Z_1$$

$$Z = Z_1 Z_2 \quad T = U_1 + U_2 \quad M = S_1 + S_2$$

$$V = (-1)^\delta (U_1 - U_2) \qquad N = (-1)^\delta (S_1 - S_2)$$

$$E = M + V \quad F = ZE \quad L = FE$$

$$R = T^2 - U_1 U_2 + Z(aZ + N) \quad W = R^2 - G$$

Projective form of [BDJ04] formula

- Let $x_i = X_i/Z_i, y_i = Y_i/Z_i$. Then

$$X_3 = 2FW \quad Y_3 = R(G - 2W) - LFM \quad Z_3 = 2F^3$$

$$U_1 = X_1 Z_2 \qquad U_2 = X_2 Z_1$$

$$S_1 = Y_1 Z_2 \qquad S_2 = Y_2 Z_1$$

$$Z = Z_1 Z_2 \quad T = U_1 + U_2 \quad M = S_1 + S_2$$

$$\boxed{V = (-1)^\delta (U_1 - U_2)} \qquad \boxed{N = (-1)^\delta (S_1 - S_2)}$$

$$E = M + V \quad F = ZE \quad L = FE$$

$$R = T^2 - U_1 U_2 + Z(aZ + N) \quad W = R^2 - G$$

- If a conditional addition occurs in either $\boxed{U_1 - U_2}$ or $\boxed{S_1 - S_2}$, then it is not a point doubling.

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:
  1. conditional subtraction: one of $U_1 = X_1 Z_2$, $U_2 = X_2 Z_1$ but not the other

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:
  1. conditional subtraction: one of $U_1 = X_1 Z_2$, $U_2 = X_2 Z_1$ but not the other
  2. conditional subtraction: one of $S_1 = Y_1 Z_2$, $S_2 = Y_2 Z_1$ but not the other

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:
  1. conditional subtraction: one of $U_1 = X_1Z_2$, $U_2 = X_2Z_1$ but not the other
  2. conditional subtraction: one of $S_1 = Y_1Z_2$, $S_2 = Y_2Z_1$ but not the other
  3. conditional addition: $V = (-1)^\delta(U_1 - U_2)$

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:
  1. conditional subtraction: one of $U_1 = X_1 Z_2$, $U_2 = X_2 Z_1$ but not the other
  2. conditional subtraction: one of $S_1 = Y_1 Z_2$, $S_2 = Y_2 Z_1$ but not the other
  3. conditional addition: $V = (-1)^{\delta}(U_1 - U_2)$
  4. conditional addition: $N = (-1)^{\delta}(S_1 - S_2)$

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:
  1. conditional subtraction: one of $U_1 = X_1 Z_2$, $U_2 = X_2 Z_1$ but not the other (prob.: $p_{\mathrm{diff}} \approx 3/8$)
  2. conditional subtraction: one of $S_1 = Y_1 Z_2$, $S_2 = Y_2 Z_1$ but not the other (prob.: $p_{\mathrm{diff}} \approx 3/8$)
  3. conditional addition: $V = (-1)^\delta (U_1 - U_2)$
  4. conditional addition: $N = (-1)^\delta (S_1 - S_2)$

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:
  1. conditional subtraction: one of $U_1 = X_1 Z_2$, $U_2 = X_2 Z_1$ but not the other (prob.: $p_{\text{diff}} \approx 3/8$)
  2. conditional subtraction: one of $S_1 = Y_1 Z_2$, $S_2 = Y_2 Z_1$ but not the other (prob.: $p_{\text{diff}} \approx 3/8$)
  3. conditional addition: $V = (-1)^{\delta}(U_1 - U_2)$ (prob.: $p_{\text{add}} \approx \frac{1}{2}$)
  4. conditional addition: $N = (-1)^{\delta}(S_1 - S_2)$ (prob.: $p_{\text{add}} \approx \frac{1}{2}$)

Distinguishing point additions

- We now have four different conditional events which can distinguish a point addition from a point doubling:
  1. conditional subtraction: one of $U_1 = X_1 Z_2$, $U_2 = X_2 Z_1$ but not the other (prob.: $p_{\text{diff}} \approx 3/8$)
  2. conditional subtraction: one of $S_1 = Y_1 Z_2$, $S_2 = Y_2 Z_1$ but not the other (prob.: $p_{\text{diff}} \approx 3/8$)
  3. conditional addition: $V = (-1)^{\delta}(U_1 - U_2)$ (prob.: $p_{\text{add}} \approx \frac{1}{2}$)
  4. conditional addition: $N = (-1)^{\delta}(S_1 - S_2)$ (prob.: $p_{\text{add}} \approx \frac{1}{2}$)
- Under the assumption that these events occur independently, the probability of detecting a point addition when it occurs is

$$p_{\text{dist}} = 1 - (1 - p_{\text{diff}})^2 (1 - p_{\text{add}})^2 \approx \frac{9}{10} \ .$$

Number of unidentified additions

- The number of unidentified additions in a key of length n follows a binomial distribution with success probability

$$\frac{1 - p_{\text{dist}}}{2} \approx \frac{1}{20} \ .$$

Number of unidentified additions

- The number of unidentified additions in a key of length n follows a binomial distribution with success probability

$$\frac{1-p_{\mathrm{dist}}}{2} \approx \frac{1}{20} \ .$$

- From this we get the expected number of missing additions and the probability that at most k additions are not identified.

Analysis of attack for multiple curve sizes

To compare with [Wal04], we look at the best (in terms of unidentified point additions) of 512 random sample traces of a point multiplication.

field size (bits):	192	256	384	521
[Wal04]'s attack:				
expected missing additions:	19.2	26.6	41.5	57.9
search space:	$2^{17.6}$	$2^{30.4}$	$2^{56.0}$	$2^{84.2}$
our attack:				
expected missing additions:	2	4	8	13
search space:	$2^{9.67}$	$2^{18.9}$	$2^{37.2}$	$2^{59.4}$

Analysis of attack for multiple curve sizes

Instead of looking at the best of 512 random sample traces, consider how many traces are required to obtain a trace with at most 3 unidentified point additions.

Analysis of attack for multiple curve sizes

Instead of looking at the best of 512 random sample traces, consider how many traces are required to obtain a trace with at most 3 unidentified point additions.

field size (bits):	192	256	384	521
expected number of traces:	67	746	$2^{17.1}$	$2^{25.2}$
search space:	$2^{20.1}$	$2^{21.4}$	$2^{23.1}$	$2^{24.4}$

Replace reduction $\mod p$ with constant-time operation

- Assume $a, b, c \in \{0, \ldots, p-1\}$.
- Precompute multiples mp, $m \in \{1, 2\}$.

Replace reduction $\mod p$ with constant-time operation

- Assume $a, b, c \in \{0, \ldots, p-1\}$.
- Precompute multiples mp, $m \in \{1, 2\}$.
- Modular subtraction: $a - b \mod p$ becomes $(2p + a - b) - mp$, where $m \in \{1, 2\}$ is chosen based on size of $(2p + a - b)$.

Replace reduction $\bmod p$ with constant-time operation

- Assume $a, b, c \in \{0, \ldots, p-1\}$.
- Precompute multiples mp, $m \in \{1, 2\}$.
- Modular subtraction: $a - b \mod p$ becomes $(2p + a - b) - mp$, where $m \in \{1, 2\}$ is chosen based on size of $(2p + a - b)$.
- Modular addition: $a + b \mod p$ becomes $(a + b + p) - mp$, where $m \in \{1, 2\}$ is chosen based on size of $(a + b + p)$.
- Montgomery reduction: $c \mod p$ becomes $(c + p) - mp$, where $m \in \{1, 2\}$ is chosen based on size of $(c + p)$.

Separate operations and modular reductions

- Assume $a, b \in \{0, \ldots, p-1\}$.
- $a + b$ is at most $2p - 2$, which is only 1 bit longer than $p - 1$, so we don't need to reduce right away.
- Modify operations performed later (e.g., Montgomery multiplication) to have a larger domain.
- Example: Reduction in Montgomery modular multiplication allowed to return outputs between 0 and $2p - 1$, and allowed to accept products between 0 and $16p^2$.

Conclusions

- [BDJ04] provide an infinite family of unified point addition formulæ that unify the sequence of field operations. The implementation of these field operations still matters.
- Our extension of the attack of [Wal04] demonstrates that careful attention must be paid to the implementation of field operations: every type of field operation should have constant runtime.

References

[BDJ04] E. Brier, I. Déchène, and M. Joye.
Unified point addition formulæ for elliptic curve cryptosystems.
In *Embedded Cryptographic Hardware: Methodologies and Architectures*, Nova Science Publishers, 2004.

[BJ02] E. Brier and M. Joye.
Weierstraß elliptic curves and side-channel attacks.
In *Public Key Cryptography – PKC 2002*, LNCS 2274:87–100, Springer-Verlag, 2002.

[Joy05] M. Joye.
Defences against side-channel analysis.
In *Advances in Elliptic Curve Cryptography*, pp. 87–100, Cambridge University Press, 2005.

[Wal04] C. D. Walter.
Simple power analysis of unified code for ECC double and add.
In *CHES 2004*, LNCS 3156:191–204, Springer-Verlag, 2004.