Projective Coordinates Leak

David Naccache, (Gemplus)

Nigel Smart, (U. Bristol)

J. Stern, (ENS)

A new kind of side-channel

Side channel's often thought of as physical

- Power analysis
- Timing analysis
- EMF

We show a type of software/data side channel

- The side channel depends on the representation of data

Two representations

- Equivalent mathematically
- But one leaks information

Elliptic Curve Points

In this talk restrict to curves mod p

- Easily generalise to other curves

Consider the curve

$$E : Y^2 = X^3 + aX + b$$

In affine coordinates a point is represented by a pair

- (x, y)

For each group element there is exactly one representative in affine coordinates.

Elliptic Curve Points

Affine coordinates have problems:

- Computation with affine coordinates is expensive
- Division is slow
- Often to aid computation one uses projective coordinates

Usually

- Perform computation using projective coordinates
- Convert back to affine at end of protocol
  - Still requires one final division operation

Projective Points

In talk we restrict to Jacobian projective coordinates

- Easily generalise to other coordinate systems

Represent $P = (x, y)$ by $P = (X, Y, Z)$ where

- $x = X/Z^2$
- $y = Y/Z^3$
- Z is non-zero

Hence for each group element there are $p-1$ representatives in projective coordinates

- One for every non-zero value of Z.

A Lazy Card/Device

To avoid needing to implement a division operation one could imagine a variant of Diffie–Hellman.

- The card is used to compute Diffie–Hellman session keys for a user.

Suppose a card has a static DH key pair $(a, [a]P)$

- It takes an input point Q
- Computes $(X, Y, Z) \leftarrow [a]Q$
- Outputs the projective representation (X, Y, Z)

The owner then converts this back to affine coordinates to obtain the Diffie–Hellman secret.

- Conversion to affine occurs off the card.
- We shall see that this will leak some of the bits of k.

Possible Lazy Signature Protocol

Consider the following similar signature scheme

Keys

- x and $Q \leftarrow [x]G$.

Sign

- Pick $k \in_R \{1, \ldots, r\}$
- Compute $(X, Y, Z) \leftarrow [k]G$
- Compute $s \leftarrow k - xH(m, X, Y, Z) \pmod r$
- Output (X, Y, Z, s)

Verify

- Compute $P \leftarrow [d]G + [H(m, X, Y, Z)]Q$
- If $P \neq \text{Affine}(X, Y, Z)$ reject

Using techniques of Howgrave-Graham, Smart, Nguyen, Shparlinski if some bits of k are leaked for enough signatures then can recover x.

- We shall see that some bits are leaked.

Problem

Consider the binary exponentiation algorithm for $Q = [k]G$.

- $Q \leftarrow O$
- For $j = l - 1$ downto 0
  - $Q \leftarrow [2]Q$
  - If $k_j = 1$ then $Q \leftarrow Q + G$
- Return Q

Suppose all calculations are performed using projective coordinates

- G is held in affine form.

Question

- Does the projective representation of the final Q reveal whether the final bit was zero or not ?
  - This is a possibility since the projective representation is redundant

Projective Sets

For an affine point on an elliptic curve $P = (x, y)$ let

$$S_P = \{(\lambda^2 x, \lambda^3 y, \lambda) : \lambda \in \mathbb{F}_q^*\}.$$

Hence S_P denotes the set of all equivalent projective representations of P.

Given affine G we can define a map of sets

$$\psi_{P,P+G} : S_P \longrightarrow S_{P+G}$$

corresponding to the exact addition formulae used.

Similarly one can define a map for doubling

$$\psi_{P,[2]P} : S_P \longrightarrow S_{[2]P}.$$

Projective Sets

$$\psi_{P,P+G} : S_P \longrightarrow S_{P+G}$$

Our previous question now becomes

- Given an element of S_{P+G} and G can we tell whether it has resulted in an application of $\psi_{P,P+G}$?

In other words

- Is $\psi_{P,P+G}$ surjective ?

Projective Sets

It is easy, by studying the standard addition formulae, to deduce that the following holds, for Jacobian projective coordinates in large prime characteristics:

$$\begin{array}{llll}
\text{If } q \equiv 1 \mod 3 & \text{then} & \psi_{P,P+G} & \text{is a } 3 \rightsquigarrow 1 \text{ map.} \\
\text{If } q \equiv 2 \mod 3 & \text{then} & \psi_{P,P+G} & \text{is a } 1 \rightsquigarrow 1 \text{ map.} \\
\text{If } q \equiv 1 \mod 4 & \text{then} & \psi_{P,[2]P} & \text{is a } 4 \rightsquigarrow 1 \text{ map.} \\
\text{If } q \equiv 3 \mod 4 & \text{then} & \psi_{P,[2]P} & \text{is a } 2 \rightsquigarrow 1 \text{ map.}
\end{array}$$

Moreover, given an element in the codomain it is easy to determine all of its preimages if it has any.

Backtracking Algorithm

This gives us the following backtracking algorithm:

- Given $Q = [k]G$ in projective coordinates
- See if $Q \in \mathrm{Im}(\psi_{P,P+G})$
  - If it is compute all preimages P
  - If not set $P = Q$
- See if $P \in \mathrm{Im}(\psi_{P,[2]P})$
  - If it is compute all preimages P
  - If not backtrack
- Repeat for the next bit

Backtracking Algorithm

Problem is that the number of cases explodes

- Hence, always backtrack after 5 bits (say) (but keep guess).

In many cases after exploring all possibilities for the first 5 bits we will actually know the trailing bit.

In other cases have a pretty good idea but not definite information

In other cases really do not know

- Too many paths have been created in the execution tree.
- Not enough pruning been done

Experiments:- Binary Exponentiation

We ran some experiments using the above backtracking method and obtained the following probabilities:

$q \mod 12$	1	5	7	11
$\Pr$[parity determined$\|k$ even]	0.98	0.71	0.80	0.50
$\Pr$[parity determined$\|k$ odd]	0.95	0.74	0.50	0.47
$\Pr$[parity determined]	0.96	0.72	0.65	0.48

Experiments:- Signed Sliding Window

A similar algorithm can be run on any exponentiation algorithm

- e.g. signed sliding window method with window width 5...

$q \mod 12$	1	5	7	11
$\Pr$[parity determined$\vert k$ even]	0.86	0.00	0.05	0.00
$\Pr$[parity determined$\vert k$ odd]	0.81	0.75	0.49	0.53
$\Pr$[parity determined]	0.81	0.37	0.27	0.26
$\Pr[k \mod 32$ determined]	0.42	0.01	0.01	0.00

Protections

As a protection one should

Either

- Only ever transport affine coordinates

Or

- Randomize projective coordinates before transmission

$$(X, Y, Z) \longrightarrow (\lambda^2 X, \lambda^3 Y, \lambda Z)$$

Conclusion

We have shown how use of transmitted projective coordinates can leak information

Hence, representation of elliptic curve points is important

- Issues related to black-box-group assumption in some security proofs.

Note: Internal use of projective coordinates is no security risk.