Side-channel attacks on PKC

Lejla Batina

Institute for Computing and Information Sciences – Digital Security
Radboud University Nijmegen

4th June 2015

Radboud University Nijmegen

Outline

What SPA adversary can

- Sometimes even recover the key from one (or a few traces)

What SPA adversary can

- Sometimes even recover the key from one (or a few traces)
- Exploit new attack techniques

What SPA adversary can

- Sometimes even recover the key from one (or a few traces)
- Exploit new attack techniques
  $\rightarrow$ Online Template Attacks

What SPA adversary can

- Sometimes even recover the key from one (or a few traces)
- Exploit new attack techniques
  $\rightarrow$ Online Template Attacks
- Defeat some countermeasures such as

What SPA adversary can

- Sometimes even recover the key from one (or a few traces)
- Exploit new attack techniques
  $\rightarrow$ Online Template Attacks
- Defeat some countermeasures such as
  $\rightarrow$ scalar randomization

What SPA adversary can

- Sometimes even recover the key from one (or a few traces)
- Exploit new attack techniques
  $\rightarrow$ Online Template Attacks
- Defeat some countermeasures such as
  $\rightarrow$ scalar randomization
- Challenge: new horizontal attacks belong to SPA techniques

SPA-resistant modular exponentiation

Square-and-multiply always

Input: x, $d = (d_{t-1}, d_{t-2}, ..., k_0)_2$
Output: $y = x^d \mod N$

1: $R_0 \leftarrow 1, R_1 \leftarrow 1, R_2 \leftarrow x$
2: **for** $i = t-1$ downto 0 **do**
3: $\quad R_0 \leftarrow R_0^2 \mod N$
4: $\quad b \leftarrow 1 - d_i; R_b \leftarrow R_b \cdot R_2 \mod N$
5: **end for**
6: **return** R_0

When $d_i = 0$ there is a dummy multiplication!

DPA-resistant modular exponentiation

Randomizing message

Input: $m, d, N,$
Output: $c = m^d \mod N$
1: $r = Random()$
2: $m_s \leftarrow rm$
3: $v \leftarrow {m_s}^d \mod N$
4: $u \leftarrow r^d \mod N$
5: $c \leftarrow \frac{v}{u} \mod N$
6: **return** c

DPA-resistant modular exponentiation

Randomizing exponent

Input: $m, d, N, \phi(N)$,
Output: $c = m^d \mod N$

1: $r = Random()$
2: $d' \leftarrow d + r\phi(N)$
3: $c \leftarrow m^{d'} \mod N$
4: **return** c

ECDLP and scalar multiplication

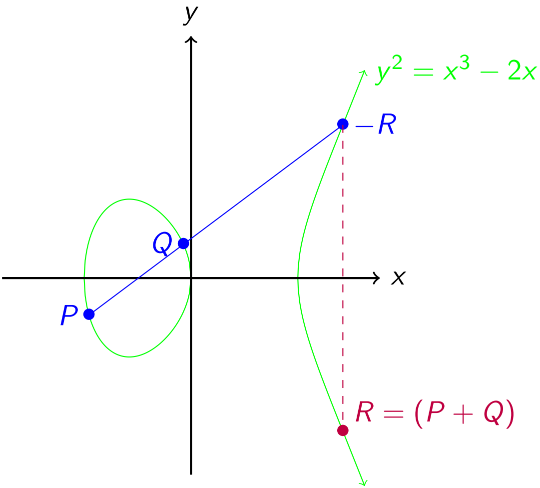
ECDLP

Let E be an elliptic curve over a finite field $\mathbb{F}_q$, $G =< P >$ a cyclic subgroup of $E(\mathbb{F}_q)$ and $Q \in G$. ECDLP is the problem of finding $k \in \mathbb{Z}$ such that $Q = kP$

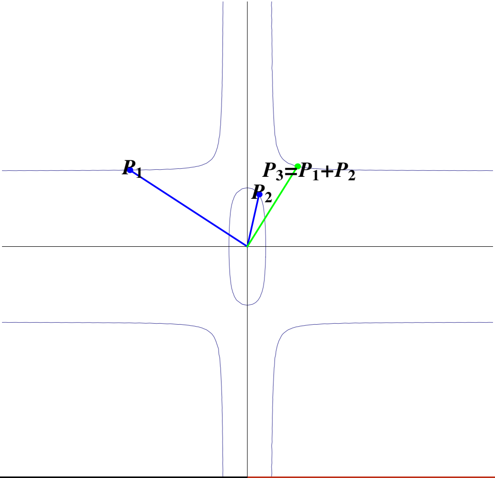
The scalar multiplication kP is the crucial computation in ECC.

$$kP = \underbrace{P + P + ... + P}_{k\text{-times}}$$

Addition rule for Weierstrass equation: $E : y^2 = x^3 - 2x$



Addition law on twisted Edwards curves $E_d : x^2 + y^2 = 1 + dx^2y^2$ defined over a field K, with characteristic $\neq 2$ and $d \in K \setminus \{0, 1\}$

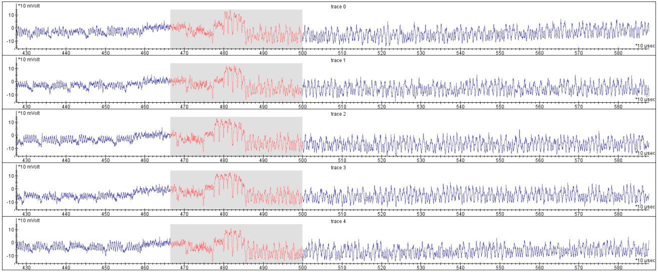


Attacks on ECC

- Simple SPA attacks can be counteracted by a balanced scalar multiplication algorithm.
- The choice of attacks varies for different protocols e.g. the protocol determines scenario.
  Example: Attacks on ECDSA are attacks on (modular) multiplication or on modular multiplication.

SPA on ECC scalar multiplication

5 traces of the first round of Limm-Lee algorithm. Pattern: 11001



Slide credit: L. Chmielewski.

Attacking Schnorr identification protocol

Table 1. Schnorr Identification Protocol

Prover		Verifier
$r \in_R \mathbb{Z}_n$		
$X \leftarrow rP$	$\xrightarrow{X}$	
	$\xleftarrow{e}$	$e \in_R Z_{2^t}$
$y = ae + r$	$\xrightarrow{y}$	
		if $yP + eZ = X$ then accept

- SPA on rP might reveal r. But, is knowing r useful?

Attacking Schnorr identification protocol

Table 1. Schnorr Identification Protocol

Prover		Verifier
$r \in_R \mathbb{Z}_n$		
$X \leftarrow rP$	$\xrightarrow{X}$	
	$\xleftarrow{e}$	$e \in_R Z_{2^t}$
$y = ae + r$	$\xrightarrow{y}$	
		if $yP + eZ = X$ then accept

- SPA on rP might reveal r. But, is knowing r useful? Yes, if r is known, compute $a = (y - r)e^{-1}$
- If group ops are SPA resistant, try DPA on points and recover key bit-by-bit.

Attacking Schnorr identification protocol

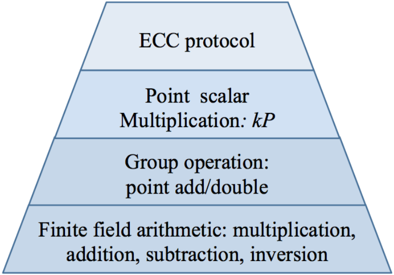
Table 1. Schnorr Identification Protocol

Prover		Verifier
$r \in_R \mathbb{Z}_n$		
$X \leftarrow rP$	$\xrightarrow{X}$	
	$\xleftarrow{e}$	$e \in_R Z_{2^t}$
$y = ae + r$	$\xrightarrow{y}$	
		if $yP + eZ = X$ then accept

- SPA on rP might reveal r. But, is knowing r useful?
  Yes, if r is known, compute $a = (y - r)e^{-1}$
- If group ops are SPA resistant, try DPA on points and recover key bit-by-bit.
  Toy example: $3P$ is only computed iff second key bit equals 1.

What do we want from countermeasures?

- Countermeasures can be applied on all levels of the hierarchy
- One should make sure that leaked information is useless



ECC countermeasures

- Protocol level: leakage-aware protocol design
- Scalar-mult level: random scalar-splitting, randomize scalar and points (by other points)
- Special scalar-indistinguishable group operations: double-and-add alway, add always, Montgomery
- Randomize intermediate results: projective coordinates
- Secure hardware, randomization

Template Attacks

- Combination of statistical modeling and power-analysis attacks
- Template-Building Phase
- Template-Matching Phase

Template Attacks

- Combination of statistical modeling and power-analysis attacks
- Template-Building Phase
- Template-Matching Phase
- Messerges, Dabbish, Sloan [1999]
  - MESD attack requires the attacker to run about 200 trial exponentiations for each bit of the secret exponent
- Medwed and Oswald [2008]
  - Offline DPA attack on EC scalar multiplication
  - Covariance matrix and mean values of pairs (d_i, k_j) of interesting points
  - Template-traces for 50 intermediate values per key-bit

Main ideas behind Online Template Attacks

- OTA: One full target trace and one template trace per key-bit are enough to recover the secret scalar.
- Focus on key dependent assignments within scalar multiplication.
- A variant of multiple-shot SPA, combining techniques from horizontal-collision and template attacks.

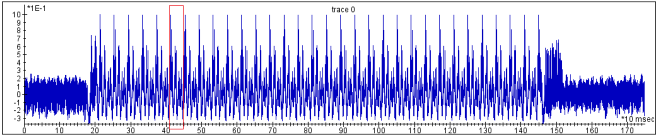


Figure: Target trace: 32 rounds of scalar multiplication for Edwards curve

Advantages of the technique

- No cumbersome pre-processing template building
- No previous knowledge of the leakage model

Advantages of the technique

- No cumbersome pre-processing template building
- No previous knowledge of the leakage model
- Works against scalar randomization and changing point representation
- Works against SPA and some DPA protected implementations

Advantages of the technique

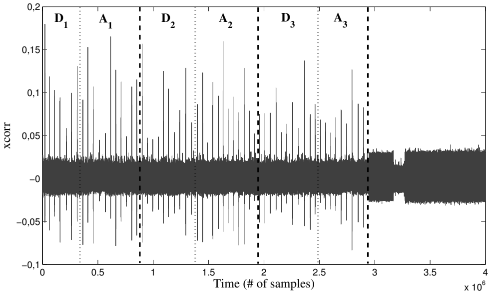
- No cumbersome pre-processing template building
- No previous knowledge of the leakage model
- Works against scalar randomization and changing point representation
- Works against SPA and some DPA protected implementations
- Applicable to Montgomery ladder and constant-time implementations
- Experimentally confirmed on the twisted Edwards curve used in Ed25519 signature scheme, Brainpool BP256r1 curve and NIST SecP256r1 curve

Attack assumptions

1. The attacker obtains only *1 target trace*. He may obtain several *template traces per key-bit*.
   (For PA: 1 template trace, for EM: 10 template traces)
2. Template traces are generated *after* obtaining the target trace, i.e. "online" or "on-the-fly".
3. Template traces are obtained on the target device or a similar device with *limited control* over it.
4. The attacker can change input points in the similar device.
5. No branches in algorithm, but at least *one key-dependent assignment*.

Attack methodology: 1.Profiling of the device

- Acquire a full target trace during execution of scalar multiplication.
- Locate the doubling and addition performed at each round.
- Find multiples mP of the input point P.



Attack methodology: 2.Template Matching

- Obtain template traces with mP, m is chosen according to the algorithm used in the target device.
- Correlate the output of $(i+1)$-iteration of target trace with input of i-iteration of template trace for each scalar bit (for unblinded scalar).

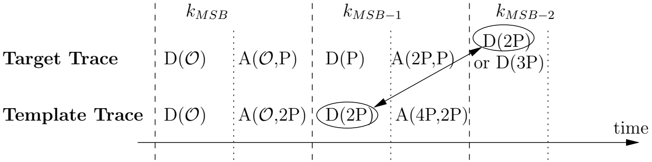


Figure: Correlation of $(i+1)$-iteration of target with i-iteration of template

OTA on double-and-add-always

Optimized double-add-always on twisted Edwards curve

Input: P, $k = (k_{x-1}, k_{x-2}, \ldots, k_0)_2$
Output: $Q = kP$

1: $R_0 \leftarrow P$
2: **for** $i = x - 2$ downto 0 **do**
3: $\quad R_0 \leftarrow 2R_0$
4: $\quad R_1 \leftarrow R_0 + P$
5: $\quad R_0 \leftarrow R_{k_i}$
6: **end for**
7: **return** R_0

$k = 100$ $R_0 = P$ $R_0 = 2P, R_1 = 3P$, return $2P$ $R_0 = 4P, R_1 = 5P$, return $4P$
$k = 110$ $R_0 = P$ $R_0 = 2P, R_1 = 3P$, return $3P$ $R_0 = 6P, R_1 = 7P$, return $7P$

OTA on Montgomery Ladder

Montgomery ladder
on twisted Edwards curve

Input: P,
$k = (k_{x-1}, k_{x-2}, ..., k_0)_2$
Output: $Q = k \cdot P$

```
1: R_0 ← P
2: R_1 ← 2 · P
3: for i = x − 2 downto 0 do
4:     b = 1 − k_i
5:     R_b = R_0 + R_1
6:     R_{k_i} = 2 · R_{k_i}
7: end for
8: return R_0
```

$k = 100$
$R_0 = P, R_1 = 2P$
$b = 1 \quad R_1 = 3P, R_0 = 2P$, return $2P$
$b = 1 \quad R_1 = 5P, R_0 = 4P$, return $4P$
$k = 110$
$R_0 = P, R_1 = 2P$
$b = 0 \quad R_0 = 3P, R_1 = 4P$, return $3P$
$b = 1 \quad R_1 = 7P, R_0 = 6P$, return $6P$

Setup

- ATMega163 microcontroller
- NaCl implementation of twisted Edwards curve with unified formulas
- $\mathcal{E}_p : x^2 + y^2 = 1 + dx^2y^2$, with $d = -(121665/121666)$, $p = 2^{255} - 19$
- High security level (at least 128−bits of security) and constant time implementation

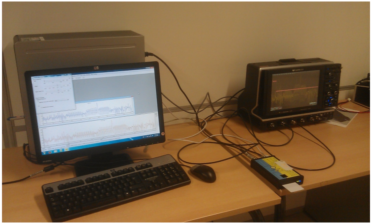


Figure: Our lab setup

OTA on twisted Edwards curve with Power Analysis

- Choose input point $P = \{P_x, P_y, P_z, P_t\}$ for the target trace.
- Compute $2P$ or $3P$ in extended coordinates with the same addition formulas.
- Correct bit assumptions have $84 - 88\%$ matching patterns, wrong bit assumptions drops to $50 - 72\%$. Pattern matching threshold: 80%.

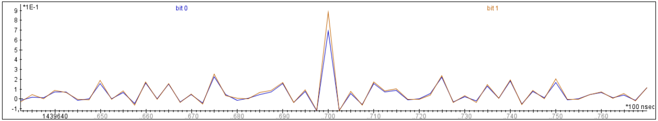


Figure: Pattern match of P on card 1 to $2P$ on card 2 (blue) and to $3P$ on card 2 (brown) for MSB of scalar 1100

[L. Batina, L. Chmielewski, L. Papachristodoulou, P. Schwabe and M. Tunstall. Online Template Attacks. In INDOCRYPT 2014 - 15th International Conference on Cryptology in India, pages 21-36, 2014.]

New results - EM Analysis

♯ traces	$\rho_{k_i} > \rho_{\neg k_i}$	$\rho_{k_i} \leq \rho_{\neg k_i}$	Success Rate
1	569	431	56,90%
10	807	193	80,70%
50	916	84	91,60%
100	998	2	99,80%

Table: Success rate for different number of traces - Vertical OTA on Brainpool

New results

Multiplication of two 32-bit words in PolarSSL.

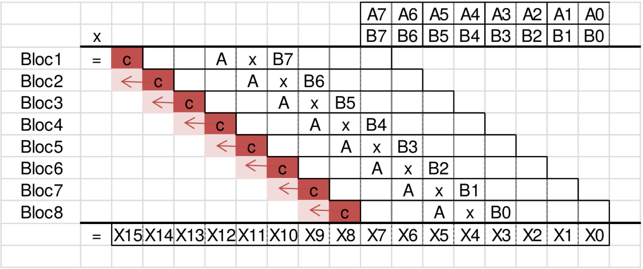


Figure: Propagation of carry during multiplication

New results

OTA on Brainpool curve with EM Analysis-Horizontal.
100% success rate with one template trace per bit.

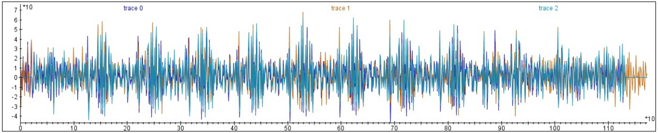


Figure: No propagation of carry

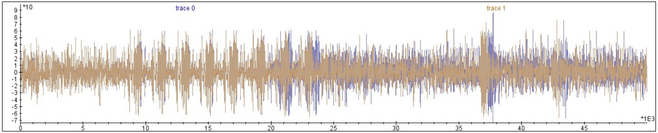


Figure: Propagation of carry

Conclusions

- Horizontal techniques including OTA are serious issues for ECC implementers
- Can countermeasures be defeated?
- Future work: implement countermeasures (randomize input point, work in isomorphic field) and try new attacks.