Side-Channel Analysis on Blinded Regular Scalar Multiplications

Benoit Feix   Mylène Roussellet   **Alexandre Venelli**

Target of our paper

- **Elliptic Curve Cryptosystems (ECC) implemented on embedded devices by industrials**
  - **Use of international standards like NIST FIPS186-2 or SEC2**
- **We are looking for their resistance against non-profiled side-channel attacks**
  - **The attacker has no access to an open device**
    - Template attacks → talk « Online Template Attacks »
  - **More restrictive from an adversary point of view, hence generally more difficult to mount on protected devices**
- **We propose an new attack path on a industrially standard implementation of scalar multiplication algorithm resistant against previously known non-profiled attacks**

Target of our paper

- **Example of targeted implementation :**
  - **Elliptic curve NIST P-192**
  - **SSCA-resistance**
    - Double-and-add-always
  - **DSCA-resistance**
    - Input point blinding : randomized projective coordinates
    - Exponent blinding : add a random multiple of the curve's order

- $Q = [d]P$

Agenda

1. **Background: side-channel attacks, ECC**
2. **Attack strategy**
   1. Weakness of the scalar blinding
   2. Attack with known input
   3. Attack on a fully protected algorithm
3. **Experimental results**
4. **Countermeasures**
5. **Conclusion**

Different flavors of side-channel attacks

- **Non-profiled side-channel analysis categories :**
  - **Vertical correlation attacks**
    - The original CPA from Brier et al. CHES 2004
  - **Horizontal correlation attacks**
    - Attack against exponentiation with known inputs from Clavier et al. ICS 2010
  - **Vertical collision-correlation attacks**
    - Attack against simple first-order masked AES from Clavier et al. CHES 2011
    - Attack against multiply-always exponentiation with blinded inputs from Witteman CT-RSA 2011
  - **Horizontal collision-correlation attacks**
    - The classical Big-Mac attack from Walter CHES 2001
    - Attack against atomic implementations of ECC from Bauer et al. 2013
    - Attack against blinded exponentiations from Clavier et al. INDOCRYPT 2012

Side-channel resistant scalar multiplication

- **SSCA resistance :**
  - **Regular algorithms**
    - Montgomery ladder, double-and-add-always, Joye's double-add, co-Z algorithms
  - **Unified addition formulas**
    - Same formula used for both point addition and point doubling
    - Inefficient on standardized curves, only relevant for particular curve families : Edwards, Huff, …
  - **Atomicity**
    - The point addition and point doubling are computed using the same sequence of finite field operations, hence using dummy operations

- **DSCA resistance**
  - **Scalar blinding**
    - $d' = d + r.\#E$
    - Add a random multiple of the curve's order to the secret scalar
  - **Scalar splitting**
    - Several methods : additive, multiplicative, Euclidean
    - The most efficient, the Euclidean, consists in $d' = \lfloor d/r \rfloor . r + (d \bmod r)$
  - **Randomized projective points**
    - An affine point $P = (x, y)$ can be represented in Jacobian coordinates as $(\lambda^2 x, \lambda^3 y, \lambda)$ for any non-zero λ

Side-channel resistant scalar multiplication

- **Double-and-add-always**

Algorithm 1 Double-and-add-always

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

1: $\boldsymbol{R_0} \leftarrow \boldsymbol{O}$; $\boldsymbol{R_1} \leftarrow \boldsymbol{O}$
2: **for** $j = k-1$ **to** 0 **do**
3: $\quad \boldsymbol{R_0} \leftarrow [2]\boldsymbol{R_0}$
4: $\quad b \leftarrow d_j$; $\boldsymbol{R_{1-b}} \leftarrow \boldsymbol{R_0} + \boldsymbol{P}$
5: **end for**
6: **return** $\boldsymbol{R_0}$

- **Randomized projective points**

- **Scalar blinding**

Agenda

1. Background: side-channel attacks, ECC
2. Attack strategy
    1. Weakness of the scalar blinding
    2. Attack with known input
    3. Attack on a fully protected algorithm
3. Experimental results
4. Countermeasures
5. Conclusion

Attack in 3 steps

1. **Exploit weakness in the scalar blinding CM**
   - Vertical attack → Middle part of the scalar
2. **Recover the random used for the blinding**
   - Horizontal attack → MS part of the scalar
3. **Find the remaining bits**
   - Vertical attack → LS part of the scalar

Weakness in blinded scalars

- **A possible weakness in the scalar blinding technique has been noted by Joye, Ciet since CHES 2003**

$$d' = d + r.\#E$$

Example (secp160k1)

$p = 2^{160} - 2^{32} - 538D_{16}$ [generalized] Mersenne prime

$\#E = 01\ 00000000\ 00000000\ 0001B8FA\ 16DFAB9A\ CA16B6B3_{16}$

$\Rightarrow d^* = d + r\#E = (r)_2 \parallel d_{\ell-1} \cdots d_{\ell-t} \parallel$ some bits

- **Example taken from Marc Joye's slides on ECC in the presence of faults**

- **The same weakness has also been noted by Smart, Oswald, Page in IET Information Security 2008**

Weakness in blinded scalars

- **Both remark that the middle part of d' is correlated to the most significant part of d**

- **However no key recovery attack path was found. Concerns were raised about the use of scalar blinding**

- **We provide a full key recovery attack exploiting this weakness and we show the limits of this CM**

security to be free

Classification of sparse order groups

- **Hasse's theorem:**
  - $n = \#E(F_p)$ then $(\sqrt{p} - 1)^2 \leq n \leq (\sqrt{p} + 1)^2$
  - n is close to the value of p
- **NIST FIPS186-2**
  - Curves defined over the primes: $p_{192}, p_{224}, p_{256}, p_{384}, p_{521}$
  - Hence their orders are also sparse
- **3 categories of curves**
  - Type-1: the order has a large pattern of ones,
  - Type-2: the order has a large pattern of zeros,
  - Type-3: the order has a combination of large patterns of both ones and zeros

Classification of sparse order groups

- **Notation:** $1^{[a,b]}$ → **a pattern of** 1 **bits from the bit position** a **to** b**. Respectively for** $0^{[a,b]}$
- **Types of** k**-bit curve orders** n**:**
  - Type-1: $n = 1^{[k-1,a]} + x$ with $(k-1) > a$ and $0 \le x < 2^a$
  - Type-2: $n = 2^{k-1} + 0^{[k-2,a]} + x$ with $(k-2) > a$ and $0 \le x < 2^a$
  - Type-3: $n = 1^{[k-1,a]} + 0^{[a-1,b]} + 1^{[b-1,c]} + x$ with $(k-1) > a > b > c$ and $0 \le x < 2^c$
- **Examples with standard curves:**
  - Type-1: $n = 1^{[191,96]} + x$ (NIST P-192)
  - Type-2: $n = 2^{225} + 0^{[224,114]} + x$ (SECP224k1)
  - Type-3: $n = 1^{[255,224]} + 0^{[223,192]} + 1^{[191,128]} + x$ (NIST P-256)

Random multiple of the order

- $r \in [1,2^m - 1]$ **an** m**-bit random used for the scalar blinding**
- **Representations of** $r.n$ **:**
  - Type-1: $\boldsymbol{r.n = \widetilde{r_1}.2^k + 1^{[k-1,a+m]} + x}$
  - Type-2: $\boldsymbol{r.n = r.2^k + 0^{[k-1,a+m]} + x}$
  - Type-3: $\boldsymbol{r.n = \widetilde{r_1}.2^k + 1^{[k-1,a+m]} + \widetilde{r_0}.2^{a+m} + 0^{[a-1+m,b+m]} + \widetilde{r_1}.2^{b+m} + 1^{[b-1+m,c+m]} + x}$
- **The patterns of zeros and ones are reduced by** m **bits**
- **The values** $\widetilde{r_1}$ **and** $\widetilde{r_0}$ **are directly related to** r **and** m
  - See paper for details

Adding the scalar to the random mask

- **Representations of d' with the 3 types :**
  - **Type-1:** $\mathrm{d}' = (\widetilde{r_1} + 1).2^k + d^{[k-1,a+m]} + x$ **Non-masked**
  - **Type-2:** $\mathrm{d}' = r.2^k + d^{[k-1,a+m]} + x$
  - **Type-3:** $\mathrm{d}' = (\widetilde{r_1} + 1).2^k + d^{[k-1,a+m]} + \widetilde{r_0}.2^{a+m} + d^{[a-1+m,b+m]} + (\widetilde{r_1} + 1).2^{b+m} + d^{[b-1+m,c+m]} + x$

- **We clearly distinguish the non-masked part of d'**

Agenda

1. Background: side-channel attacks, ECC
2. Attack strategy
    1. Weakness of the scalar blinding
    2. Attack with known input
    3. Attack on a fully protected algorithm
3. Experimental results
4. Countermeasures
5. Conclusion

Attack on a blinded scalar multiplication with known input

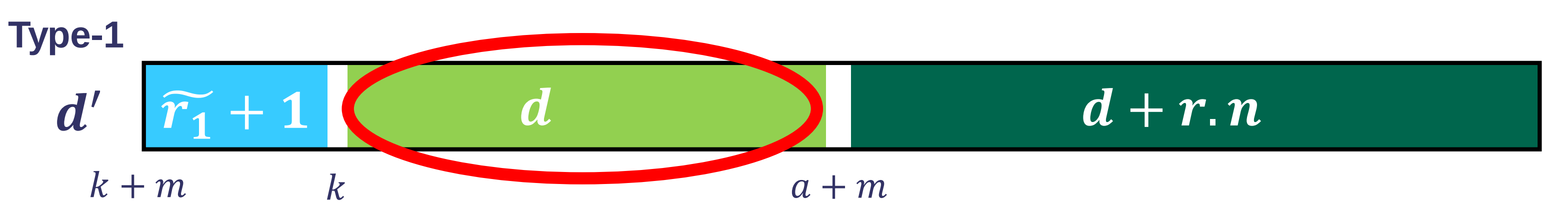
- **First, simpler scenario, the input point is known, i.e. not masked**
- **Notations:** $\{C^{(1)}, \dots, C^{(N)}\}$ **be** N **side-channel traces corresponding to the computations** $[d'^{(i)}]P^{(i)}$ **where** $d'^{(i)} = d + r^{(i)}.n$
- **We consider random factors** $r^{(i)} \in [1,2^{m-1}]$

Attack step 1

- **Goal: find the non-masked part of d'**

- **Let δ be the bit-length of this non-masked part noted $\bar{d} = d^{[a,b]}$ with $\delta = (a - b)$**

- **Most significant part of d' unknown**
  - → Vertical collision-correlation



Attack step 1

- **Collision in the double-and-add-always**

- **If** $d_j = 0$
  - $\boldsymbol{R_0} \leftarrow [2]\boldsymbol{R_0}$
  - $\boldsymbol{R_1} \leftarrow \boldsymbol{R_0} + \boldsymbol{P}$
  - $\boldsymbol{R_0} \leftarrow [2]\boldsymbol{R_0}$

j **turn**

collision

$(j+1)$ **turn**

Notation:
$In\big(ECADD(j)\big) = In(ECDBL(j+1))$

- **No collision if** $d_j = 1$

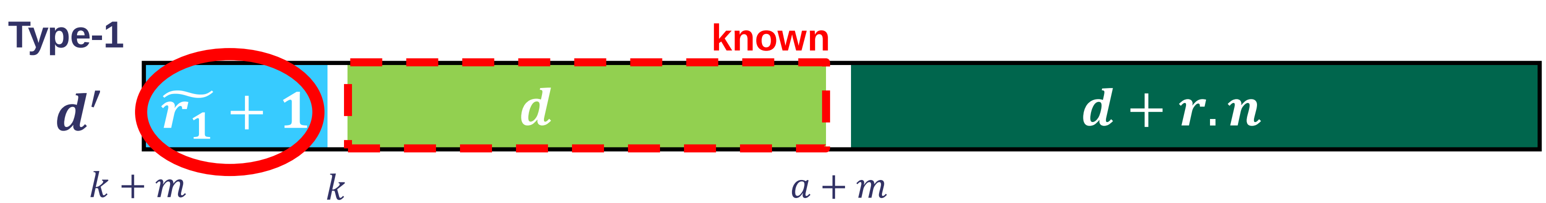
Attack step 1

- **To find $\bar{d}_j$, $0 < j < \delta$:**
  - **Let t_0 be the time sample of the side-channel trace that corresponds to $In(ECADD(j))$**
  - **Construct $\boldsymbol{\Theta_0} = \{\mathbf{C}^{(\mathbf{i})}(\mathbf{t_0})\}_{1 \le i \le N}$**
  - **Let $\boldsymbol{t_1}$ be the time sample of $In(ECDBL(j+1))$**
  - **Construct $\boldsymbol{\Theta_1} = \{\mathbf{C}^{(\mathbf{i})}(\mathbf{t_1})\}_{1 \le i \le N}$**
  - **Perform a collision-correlation $\boldsymbol{\rho}(\boldsymbol{\Theta_0}, \boldsymbol{\Theta_1})$**
    - The correlation will be maximal when $\bar{d}_j = 0$

- **For Type-3 curves, repeat the attack on all non-masked parts of d'**

Attack step 2

- **Goal: retrieve the random masks $r^{(i)}$**
- **The random values need to be retrieved from each traces $C^{(i)}, 1 \leq i \leq N$**
- **The random is present in the most significant part of the blinded scalars**
- **As the input point is known**
  - → Horizontal correlation attack

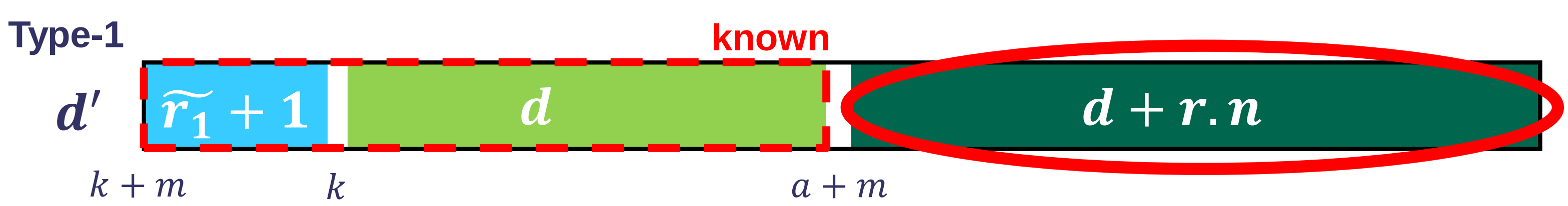


Attack step 2

- **To retrieve $r^{(i)}$:**
  - **Try all m-bit values of $r^{(i)}$**
    - A guess on $r^{(i)}$ directly gives a guess on the most significant part of $d'^{(i)}$
  - **Let $\hat{r}$ be the guess on $r^{(i)}$. It gives a sequence of elliptic curve operations that should appear at the start of $C^{(i)}$. Since $P^{(i)}$ is known, the attacker can compute the sequence and obtain $\eta = 2(m+\delta)$ intermediate points**
  - **Choose a leakage function L (e.g. Hamming weight) and compute some predicted values derived from the η points $T_j, 1 \le j \le \eta$**
  - **Construct $\Theta_1 = (\mathrm{l_j})_{1 \le \mathrm{j} \le \eta}$ with $l_j = L(T_j)$**
  - **Construct $\Theta_0 = (\mathrm{o_j})_{1 \le \mathrm{j}\eta}$ with o_j the identified points of interest related to T_j on the trace $C^{(i)}$**
  - **Compute the correlation $\rho(\Theta_0, \Theta_1)$**
    - If $\hat{r}$ is correct, maximal correlation

Attack step 3

- **Goal: recover the least significant part of d**
- **We already know**
  - The most significant bits of d (Step 1)
  - The random values $r^{(i)}, 1 \leq i \leq N$ (Step 2)
- **By guessing w unknown bits of d, we can compute guessed blinded scalars $\widehat{d'^{(i)}}$**
- **As we know the input point**
  - → Vertical correlation attack



Attack step 3

- **To find w unknown bits of d :**
  - **Guess w bits and compute the guessed blinded scalars $\widehat{d'^{(i)}}, 1 \leq i \leq N$**
  - **Choose a leakage function L**
  - **For the i-th trace, compute predicted values $l_j^{(i)} = L(T_j^{(i)})$ from the $\eta = 2w$ intermediate points $T_j^{(i)}$**
  - **Construct $\Theta_1 = \left(l_j^{(i)}\right)_{i,j}$ with $1 \leq i \leq N$ and $1 \leq j \leq \eta$**
  - **Construct $\Theta_0 = \left(\mathrm{o}_\mathrm{j}^{(\mathrm{i})}\right)_{\mathrm{i,j}}$ where $\mathrm{o}_\mathrm{j}^{(\mathrm{i})}$ is the time sample corresponding to the processing of $T_j^{(i)}$**
  - **Compute the correlation $\rho(\Theta_0, \Theta_1)$**
    - Maximal correlation when the w guessed bits are correct

1. Background: side-channel attacks, ECC
2. Attack strategy
    1. Weakness of the scalar blinding
    2. Attack with known input
    3. Attack on a fully protected algorithm
3. Experimental results
4. Countermeasures
5. Conclusion

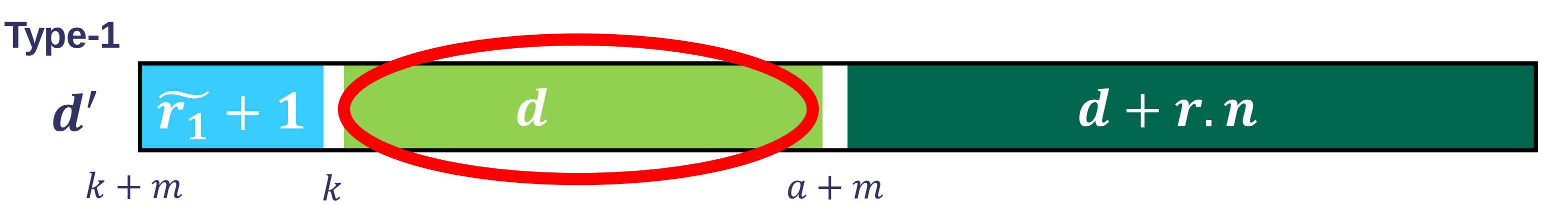
Attack on a protected scalar multiplication

- **On most state-of-the-art industrial implementations:**
  - SPA-resistant algorithm
  - DSCA protections on the scalar and the input point

- **We apply the same attack strategy in the case where the input is unknown, i.e. masked**

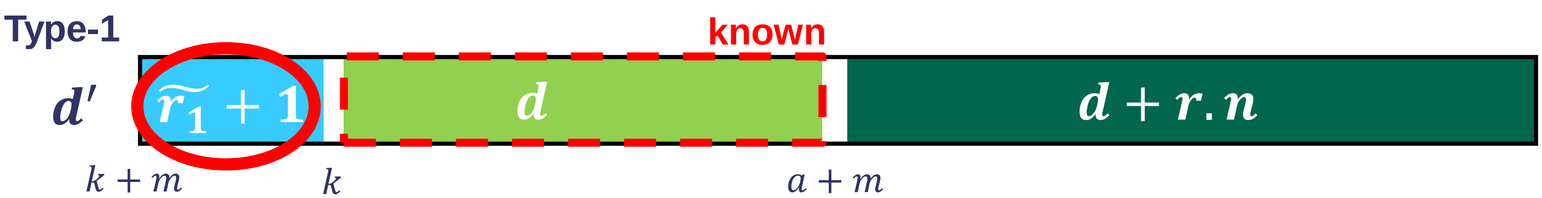
Attack step 1

- **Step 1: Vertical collision-correlation**
- **Input point not needed**
- **Same attack in the unknown input point case**



Attack step 2

- **Step 2: Horizontal correlation not possible anymore**
  - → Horizontal collision-correlation



Attack step 2

- **Collision in the double-and-add-always**

- **If** $d_j = 1$
  - $R_0 \leftarrow [2]R_0$ — j turn
  - $R_0 \leftarrow R_0 + P$ — collision
  - $R_0 \leftarrow [2]R_0$ — $(j+1)$ turn

- **If** $d_j = 0$
  - $R_0 \leftarrow [2]R_0$ — j turn
  - $R_1 \leftarrow R_0 + P$ — collision
  - $R_0 \leftarrow [2]R_0$ — $(j+1)$ turn

Attack step 2

- **To retrieve $r^{(i)}$:**
  - Try all possible m-bit values of $r^{(i)}$
  - Guessed random $\hat{r}$ → sequence of $(m+\delta)$ guessed EC operations
  - Construct $\Theta_0 = \left\{C^{(i)}\left(t_0^X(j)\right), C^{(i)}\left(t_0^Y(j)\right), C^{(i)}\left(t_0^Z(j)\right)\right\}_{1 \le j \le (m+\delta)}$ where

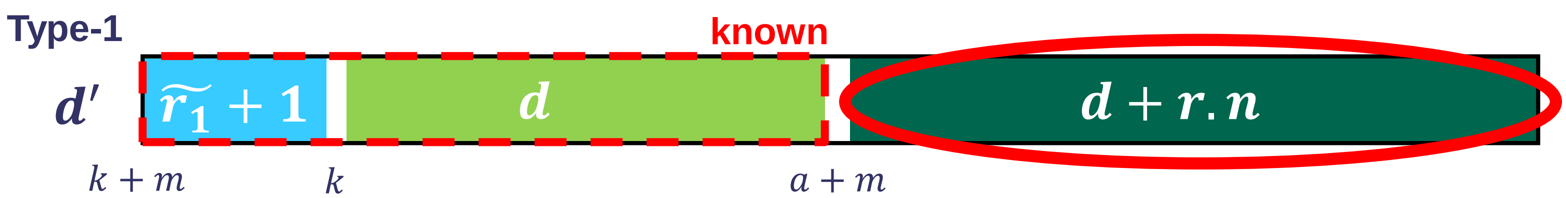
$$t_0^X(j) = \begin{cases} Out^X(ECADD(j)) \text{ if } \widehat{d'_j} = 1 \\ In^X(ECADD(j)) \text{ if } \widehat{d'_j} = 0 \end{cases}$$

  - Construct $\Theta_1 = \left\{C^{(i)}\left(t_1^X(j)\right), C^{(i)}\left(t_1^Y(j)\right), C^{(i)}\left(t_1^Z(j)\right)\right\}_{1 \le j \le (m+\delta)}$ where

$$t_1^X(j) = In^X(ECDBL(j+1))$$

  - Compute the correlation $\rho(\Theta_0, \Theta_1)$
    - Correctly guessed $\hat{r}$ gives the maximal correlation

- **Step 3: Vertical correlation not possible anymore**
  - → Vertical collision-correlation



Attack step 3

- **To find w unknown bits of d :**
  - **Guess w bits and compute the guessed blinded scalars $\widehat{d'^{(i)}}, 1 \leq i \leq N$**
  - **Construct collision vectors Θ_0 and Θ_1 similarly to the previous attack step. Consider that $u \leq \delta$ bits of d are already known, the vectors size is then $(m + u + w)N$**
  - **Compute the correlation $\rho(\Theta_0, \Theta_1)$**
    - Maximal correlation for the correctly guessed w bits

Agenda

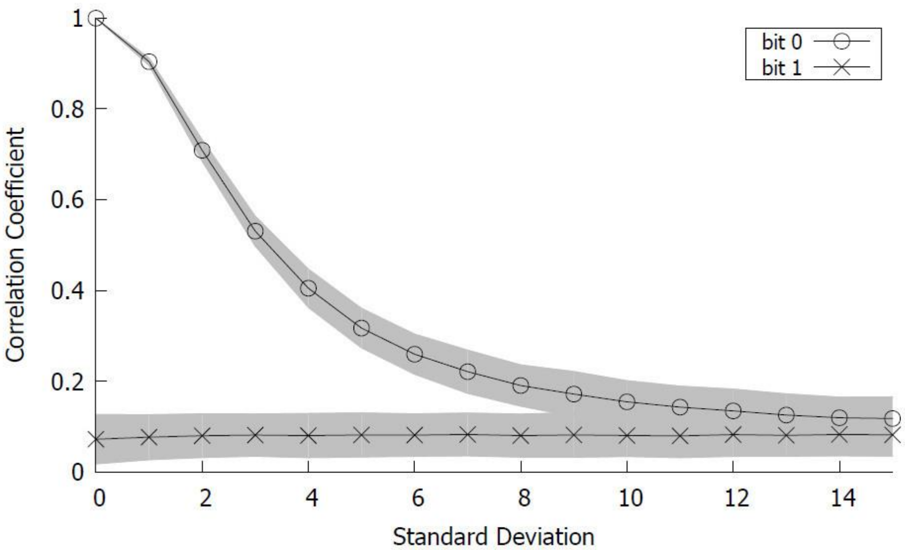
1. **Background: side-channel attacks, ECC**
2. **Attack strategy**
   1. Weakness of the scalar blinding
   2. Attack with known input
   3. Attack on a fully protected algorithm
3. **Experimental results**
4. **Countermeasures**
5. **Conclusion**

Experimentations

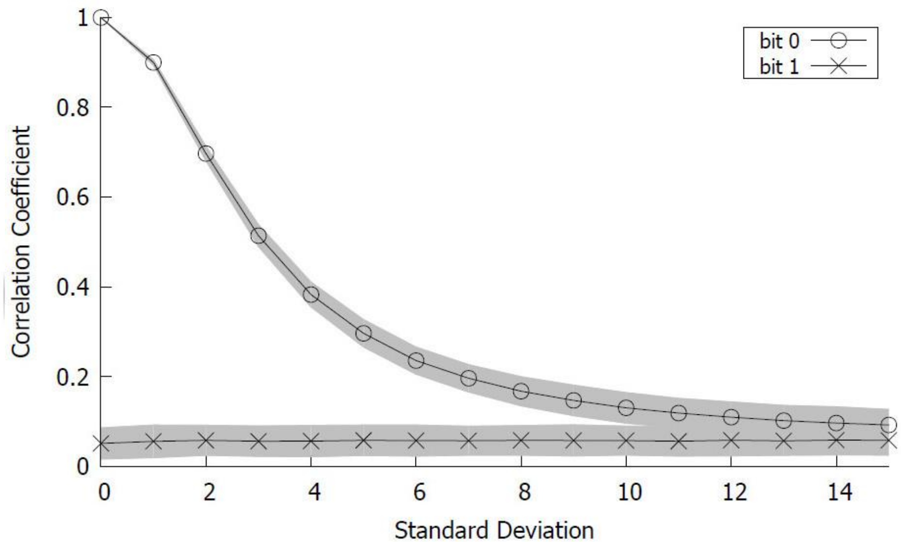
- **Simulated power traces considering the following implementation**
  - NIST P-192
  - Double-and-add-always
  - Jacobian projective coordinates with formulas add-2007-bl and dbl-2007-bl from
    - Bernstein, D.J., Lange, T.: Explicit-formulas database. http://hyperelliptic.org/EFD/g1p/auto-shortw.html
  - Random sizes of 8-bit and 16-bit to obtain reasonable computational times and to repeat our simulations for consistency
- **We consider the Hamming weight of 32-bit words as leakage model**
- **Gaussian noise with standard deviation σ is added**
- **The Pearson coefficient is used**

Simulated attack results on known input points

- **Step 1: Vertical collision-correlation**
  - **Tested using sets of 500 and 1000 traces**



500 traces



1000 traces

Simulated attack results on known input points

- **Step 2: Horizontal correlation**
  - Only need one trace
  - Success rate depends on m and σ
  - Larger random gives better results but larger computational time

- **Step 3: Vertical correlation**
  - Tested using sets of 500 and 1000 traces

Simulated attack results on known input points

- **Summary**

Attack steps	N	m	Standard Deviation σ					
			0	1	2	5	10	15
Vertical	500	-	1.0	1.0	1.0	1.0	0.88	0.74
collision-correlation	1000	-	1.0	1.0	1.0	1.0	0.99	0.76
Horizontal	-	8	1.0	1.0	1.0	1.0	1.0	0.77
correlation	-	16	1.0	1.0	1.0	1.0	1.0	0.85
Vertical	500	-	1.0	1.0	1.0	1.0	0.64	0.42
correlation	1000	-	1.0	1.0	1.0	1.0	0.84	0.52

Table 1. Success rate for known input points.

Simulated attack results on unknown input points

- **Step 1: Vertical collision-correlation**
  - Same as in the previous scenario
- **Step 2: Horizontal collision-correlation**
  - Success rate drops quicker than other attacks due to the limited number of time samples
  - Contrary to vertical attacks, this number is fixed regardless of the noise level
- **Step 3: Vertical collision-correlation**
  - Very efficient even for high σ

Simulated attack results on known input points

- **Summary**

Attack steps	N	m	Standard Deviation σ					
			0	1	2	5	10	15
Horizontal	-	8	1.0	1.0	0.9	0.1	0.02	0.01
collision-correlation	-	16	1.0	1.0	0.95	0.23	0.10	0.02
Vertical	500	-	1.0	1.0	1.0	1.0	1.0	0.97
collision-correlation	1000	-	1.0	1.0	1.0	1.0	1.0	0.99

Table 2. Success rate for unknown input points.

- **Unknown input point**
  - Full scalar recovery for noise levels up to $\sigma \approx 5$
- **Known input point**
  - Full scalar recovery for noise levels up to $\sigma \approx 10$

Agenda

1. Background: side-channel attacks, ECC
2. Attack strategy
   1. Weakness of the scalar blinding
   2. Attack with known input
   3. Attack on a fully protected algorithm
3. Experimental results
4. **Countermeasures**
5. **Conclusion**

security to be free

Countermeasures

- **Scalar splitting**
  - Euclidean splitting is the best choice
  - Often disregarded by developers as it is less efficient than scalar blinding with small random sizes
- **Scalar blinding with larger random**
  - The choice for the size m of the random depends on
    - The largest pattern size amongst all curves' order implemented
    - The maximal brute force capability of the attacker
  - Depending on this new value for m, the overhead needs to be compared to the overhead of the Euclidean splitting (1.5)
- **Atomic algorithm and unified formulas**
  - Most state-of-the-art implementations have been attacked by Bauer et al. SAC 2013

Applicability to other regular algorithms

- **Our attack paths also apply to**
  - Montgomery ladder
  - Joye's double-add
- **Only modification is on the choice of the collision variables that differs for each algorithm**
- **Does not work on the right-to-left binary algorithm lastly improved in**
  - Joye, M., Karroumi, M.: Memory-efficient fault countermeasures - Smart Card Research and Advanced Applications, 2011
- **Details in the extended version of the paper**
  - ePrint 2014/191

Agenda

1. Background: side-channel attacks, ECC
2. Attack strategy
    1. Weakness of the scalar blinding
    2. Attack with known input
    3. Attack on a fully protected algorithm
3. Experimental results
4. Countermeasures
5. **Conclusion**

Conclusion

- **We exploited a weakness in the scalar blinding to mount a full key-recovery attack on state-of-the-art protected scalar multiplications**

- **Our attack paths have good success rates even for high noise levels**
  - Known input: up to $\sigma \approx 10$
  - Unknown input: up to $\sigma \approx 5$

- **Safe solution:**
  - Any regular algorithm
  - Any input point randomization CM
  - Use Euclidean splitting as scalar randomization CM

Thanks for your attention