Twist Insecurity

Manfred Lochter[⋆] and Andreas Wiemers[⋆⋆]

Bundesamt für Sicherheit in der Informationstechnik (BSI), Germany

Abstract Several authors suggest that the use of twist secure Elliptic Curves automatically leads to secure implementations. We argue that even for twist secure curves a point validation has to be performed. We illustrate this with examples where the security of EC-algorithms is strongly degraded, even for twist secure curves.
We show that the usual blindig countermeasures against SCA are insufficient (actually they introduce weaknesses) if no point validation is performed, or if an attacker has access to certain intermediate points. In this case the overall security of the system is reduced to the length of the blinding parameter. We emphazise that our methods work even in the case of a very high identification error rate during the SCA-phase.

Keywords. Twist security, deterministic ECDSA, ECDH, random blinding, SCA

1 Introduction

Let

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

be an elliptic curve over a finite prime field $\mathbb{F}_p$ with $p > 3$. By the Hasse-Weil theorem E has $p + 1 + x$ points defined over $GF(p)$, where $|x| \leq 2\sqrt{p}$. The twisted[1] curve E^{Tw} then has $p + 1 - x$ points defined over $\mathbb{F}_p$. E is called twist-secure if both E and E^{Tw} are cryptographically strong[2]. As a minimum both $\#E(\mathbb{F}_p)$ and $\#E^{Tw}(\mathbb{F}_p)$ have to be almost prime. We fix a base point P_0 of prime order q in $E(\mathbb{F}_p)$.

[6] thoroughly discusses the security of twist secure curves and explains why relying on twist security may introduce dangers into ECC systems. The authors state: "Moreover twist security won't protect against potential side channel attacks where one would detect if the computation took place on the curve or its twist and gain information on some bits of the secret". Our paper affirms this statement using methods introduced in [11]. However, contrary to [11], we do not have to make assumptions on the form of p.

[3] summarizes applications of twist secure elliptic curves in cryptography and discusses fault attacks on unblinded Montgomery-Ladder implementations of ECC. Our method applys to a broader class of EC-scalar-multiplication methods.

Assume that a cryptographic mechanism computes dP for a secret scalar d and $P \in \langle P_0 \rangle$.

⋆ Manfred.Lochter@bsi.bund.de
⋆⋆ Andreas.Wiemers@bsi.bund.de

[1] More exactly: If $\#E(GF(p)) = p + 1 + x$ we consider the (GF(p)-isomorphism) class of curves E^{Tw} : $y^2 = x^3 + au^2x + bu^3$ with $\#E^{Tw}(GF(p)) = p + 1 - x$ as twist of E.

[2] The criteria for a curve being cryptographically strong may depend on the context the curve will be used in

At first glance, using twist secure curves seems to protect against some types of attacks attacks (see ([6], §1.2.3) for additional information). It is widely believed that the use of twist secure curves helps to improve security in the following situations:

- Invalid x: If only the x-coordinates of points are used an expanded x-coordinate could lead to a point Q in $E^{Tw}(\mathbb{F}_p)$ instead of $E(\mathbb{F}_p)$. If E^{Tw} is not cryptographically strong and if Q has smooth order on E^{Tw} an attacker might take advantage of this situation by providing x-coordinates of points that lie on E^{Tw}.
- Fault-Injection: During the computation of dP_0 a fault is introduced, that leads to computations on E^{Tw} instead of E. Then the same ideas as for "Invalid x" apply. For simplicity one can assume that the fault is injected immediately at the beginning of the computation.

Static multiplication with a secret value is used, for example, with static ECDH or with the decryption process of the elliptic curve integrated encryption scheme (ECIES) [9]. A proposal has been made by H. Krawczyk [5] that defines an authentication process for TLS 1.3 that does not rely on a signature. Within this context, static ECDH may take on additional importance.

Another area where our work is applicable are deterministic signatures [8], where deterministic ephemeral keys can be recovered and subsequently used to compute the long term signature key.

In real world applications the computation of dP is performed by using a blinded value $d + r_i q$ with a randomly chosen $r_i \in \{1, \ldots, 2^R\}$, where R is a system parameter. Blinding is a widely used counter method to thwart side-channel attacks on implementation that can be accessed by an attacker.

While the ECDSA does not use a static point multiplication its first step is a multiplication of the base-point P_0 by a random-number $k \in \{1, \ldots, q-1\}$. We will see that performing this computation with a point P_0' on one of the twists of E instead of P_0 can compromise the secret signature key for static variants of ECDSA (e.g. [8]) if no point-validation on kP_0 is performed. $Q := kP_0$ typically is also computed as $(k + r_i q)P_0$ using a blinded value $k + r_i q$ instead of k.

Remark 1. While Coron's original proposal [2] was to use $R = 20$ bit blinding it turned out that this length is not suffient. The length of the blinding needed also strongly depends on the structure of the underlying prime field. While for random prime fields 64 bit randomisation seem to be secure [10,1] recent work [11] shows that elliptic curves over special prime fields are much more vulnerable. Here up to $log_2(\sqrt{q})$ bit blinding can be attacked, depending on the so called *gap* of q. The success rate depends on the quality of the measurements.

The important observation is, that

$$\boxed{(d + r_i q)P_0' \neq (d + r_j q)P_0' \quad \text{on } E^{Tw} \qquad (*)}$$

since $q = ord_E(P_0)$ doesn't cancel out in the calculation on E^{Tw}. In combination with the following assumption (based on results of Side-Channel-Analysis)

$$\boxed{\begin{array}{ll}\text{With a fixed known error rate of } \varepsilon \in [0, 0.5) \text{ we can measure} & \\ \text{each individual bit of } d + r_i q \text{ correctly for every } i. & (**)\end{array}}$$

we get the following theorem:

Theorem 1. *Under the above assumptions on an side channel/fault attack on the static multiplication with a blinded secret d the security of the overall static Diffie-Hellman oracle is at most $N * 2^{R/2}$. (R = bitlength of blinding values. N = number of traces needed). The full generic security $log_2(\sqrt{q})$ of the scheme can only be achieved if the length of the blinding value equals the bit-length of q.*

Remark 2. The number of traces needed depends on ε, but is even for ε near to 0.5 small. See section 2.

Proof: Assume that we have observed N blinded multiplications on the twisted curve. By assumption we can measure each individual bit of $d + r_i q$ correctly with an error rate ε. In addition we know the results of the point multiplications $(d + r_i q)Q \in E^{Tw}(\mathbb{F}_p)$ where Q is a known point on E^{Tw} with known order ℓ.

We know $(d + r_i q)Q = dQ + r_i qQ$ and can compute $(r_i - r_j)qQ \in E^{Tw}(\mathbb{F}_p)$ As $|r_i - r_j| \leq 2^R$ the *exact* value $r_i - r_j$ can be computed with complexity $2^{R/2}$ using e.g. Pollard's kangoroo method [4].

Fix $i \in \{1, \ldots, N\}$. For noisy guesses $\widetilde{d + r_j q} = (d + r_j q) \oplus \Delta_j = d + r_j q + z_j$ compute the blinding values $r_i - r_j$ exactly. Then

$$\widetilde{d + r_j q} - (r_j - r_i)q = d + r_i q + z_j$$

is a new guess for $d + r_i q$. With other words: We are able to compute many noisy guesses *for one* value $d + r_i q$ from guesses for many *different* values $d + r_j q$. Unfortunately we do not know, whether the Hamming weight of the new guessing errors is still small. However, the difference of each pair of guesses can easily be computed:

$$(d + r_i q + z_{j1}) - (d + r_i q + z_{j2}) = z_{j1} - z_{j2}.$$

Via construction we know that z_j can be written as $z_j = \sum a_l 2^l$, where $a_l = 1$ if $(d + r_j q)_l = 0$ and $((d + r_j q) \oplus \Delta_j)_l = 1$. Similarly $a_l = -1$ if $(d + r_j q)_l = 1$ and $((d + r_j q) \oplus \Delta_j)_l = 0$.

Our goal is to find the value $D' = d + r_i q$ given all the values $\widetilde{d + r_j q} - (r_j - r_i)q$ with $1 \leq j \leq N$ (Remember: i is fixed). Then we reduce d' modulo q in order to recover the secret d. The naive approach is to use a window method where the width w of the window is determined by the error rate ε. As a rough estimate $\binom{\sum_{i=1}^{\lfloor w\varepsilon \rfloor} i}{w}$ should be less than a bound L which gives the number of possible error patterns in a window of width w that we want to handle.

We start with the most significant window for $d + r_i q$ and from the knowledge of $z_i - z_j$ can decide whether the most significant window of z_j can also come from an admissible error pattern for $d + r_j q$. If this is not the case the error pattern is not admissible. As we can compare with $N-1$ different traces we are able to identify the most significant bits of $d + r_i q$ with high probability and to proceed with the next window. (In practice one would use overlapping windows, thus taking carrys into account.) Note that we have made no assumption on the distribution of measurement errors. If the errors are not independent, or if some bits can be guessed correctly with higher probability this method can be used to start with high-probability error-patterns. ■

2 Application of the Wide Window Method

However, it turns out that we have reduced our problem to a problem that can more efficiently be solved by the Wide Window Method from [11] (see also [10], §3.6.1.) Here we assume, as in [11], that the bit-errors are independent. Whereas the original paper [11] assumes an error rate of around 10-15 percent for attacking ECC over special prime fields, our application is much more error tolerant. Even measurements with a bit-error of 48 percent suffice, as experiments show.

We set $v_j = d + r_j q$, $\alpha_j = r_j - r_1$ and $d' = d + r_1 q$. Then we have

$$v_j \oplus (d + r_j q) = v_j \oplus (d' + \alpha_j q) = \varDelta_j$$

We treat the $\varDelta_j$ as independent repetitions of a binomially distributed random variable with parameters $p = \epsilon$ and n the bit length of v_j. In particular, the probability of the sequence

$$(\varDelta_1, \varDelta_2, \ldots, \varDelta_N)$$

only depends on the sum of the Hamming weights of the $\varDelta_j$. Much like in [11] we try to find the correct d' iteratively by considering "bit-windows" of certain lengths. In each iteration, we decide for an likely candidate for $d' \bmod 2^w$ under an assumption for $d' \bmod 2^{w'}$ with $w > w'$

For an integer m let $\mathrm{hw}_m(x)$ be the Hamming weight of the m most significant bits of x.

Our algorithm – in simplified form – works as follows:

```
Algorithm: The Window Attack

1. We fix a window-size $s$. We set $w' := 0$, $w = s$ and $d' := 0$ .
2. Generate all $w$-bit candidates for $d'(\mathrm{mod} 2^w)$ by varying the most significant $(w - w')$ bits. (Here we assume to have just one candidate for $d'(\mathrm{mod}\, 2^{w'})$.) For each candidate $\tilde{d}$ compute the evaluation function

   $$S(\tilde{d}) := \sum_{j=1}^{N} \mathrm{hw}_{w-w'} \left( [(v_j (\mathrm{mod}\, 2^w)] \oplus \left[ (\tilde{d} + \alpha_j q)(\mathrm{mod}\, 2^w) \right] \right)$$

3. Decide for the candidate $d'(\mathrm{mod}\, 2^w)$ with the minimal $S(d')$.
   (i.e. $d'(\mathrm{mod}\, 2^w) := \mathrm{argmin}\, \{S(\tilde{d})\}$)
4. If $d'$ has been found (i.e. $(d' \bmod q)$ is the secret key): RETURN($d'$) and END.
5. If $w > n$: END.
6. Set $w' := w$ and increase $w$ by $s$. Goto step 2.
```

The algorithm can be improved by selecting more than one candidate in Step 3.

The running time of each iteration is $\mathcal{O}(N 2^{w-w'})$. The success probability of the algorithm depends on the parameters $N, \epsilon, s = w - w'$. For the correct choice of $d'(\mathrm{mod}\, 2^w)$ the evaluation function is again binomially distributed with parameter $p = \epsilon$. For an incorrect choice of $d'(\bmod\ 2^w)$ we treat the evaluation function as binomially distributed with parameter $p = 1/2$[3]. For large $N(w - w')$ we can approximate both distributions by the normal distribution[4]. Under these assumptions we just have to distinguish between two normal distributions with very high confidence of at least $\delta = 1 - 2^{-(w-w')}$. See also [7].

2.1 Experimental results

As a proof of concept we simulated the attack. Surprisingly, the attack was successful even for values of ϵ very near to 0.5. We set $\epsilon := 0.48$, $w - w' := 10$, and chose the bit length $n := 250$ for v_j. In this case the variances of the two normal distributions are almost equal so that we can use formula (4.34) of [7] to compute the minimum number N of measurements needed. We get $N = 1200$.

Results of our simulations are given in the next table. For each N we used 25 iterations of the algorithm stepping through the bits of v_j by windows of length 10. Furthermore,

[3] This means that for every summand

$$\mathrm{hw}_{w-w'}(\cdot) \approx \begin{cases} 0.5 \cdot (w - w') & ; \quad \text{if } \tilde{\mathrm{d}} \text{ incorrect} \\ \varepsilon(w - w') & ; \quad \text{if } \tilde{\mathrm{d}} \text{ correct} \end{cases}$$

holds.

[4] Note: Every $\mathrm{hw}_{w-w'}(\cdot)$ is the sum of $(w - w')$ distributions and so $S(\cdot)$ is the sum of $N(w - w')$ distributions.

we repeated the experiment 10 times for each N giving 250 computations of the minimal evaluation function. For each N in the table we give the number of good "ranks" where "Rank" is the rank of the correct $d(\bmod 2^w)$ compared to candidates output by the algorithm ("Rank 1" means that the correct $d(\bmod 2^w)$ was found by the algorithm.)

N	# of rank 1	# of rank 2	# of rank ≥ 3
600	64	34	152
1200	139	49	62
2400	232	13	5

Table1. Window attack: Exemplary simulation result

In addition, for $N = 2400$ the rank of the correct $d(\bmod 2^w)$ was always ≤ 3. We can interpret the table as follows: For $N = 2400$ we can expect that – on average – 2 decisions of the 25 iterations are incorrect. A closer look at the simulations shows that in this case the minimal evaluation function of this iteration tends to be somewhat larger. Therefore, a good strategy is to keep more than one candidate per iteration if the minimum $S(\tilde{d}$ of the evaluation function is larger than a certain bound. We did not elaborate on this further since the results of our simulation already demonstrate the basic behaviour of our attack.

3 Conclusion

Relying on twist security can lead to sloppy implementations of ECC that can degrade the security of the system. We have described vulnerabilities of implementations of ECC-algorithms using blinded static point multiplications that may result from unjustified trust into twist security. These algorithms include static Diffie-Hellman and deterministic ECDSA as well as the ECIES decryption.

We have given, and confirmed by experiments, an algorithm to exploit these vulnerabilities. This algorithm relies on methods that have previously been used to attack blinded point multiplications over special prime fields. In our scenario the algorithm turns out to be extremely efficient.

We conclude that twist security should not be considered as an feature that automatically increases the security of Elliptic Curve Cryptography. Contrary to common beliefs using twist secure curves can lead to insecure implementations and degrade security.

References

1. W. Schindler, K. Itoh. Exponent Blinding Does not Always Lift (Partial) SPA Resistance to Higher-Level Security. In *In: J. Lopez, G. Tsudik (eds.): Applied Cryptography and Network Security — ACNS 2011, Springer, LNCS 6715.*
2. J.-S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In *Cryptographic Hardware and Embedded Systems*, 1999.

3. P.-A. Fouque, R. Lercier, D. Réal, and F. Valette. Fault Attack on elliptic curve with Montgomery ladder implementation. In L. Breveglieri, S. Gueron, I. Koren, D. Naccache, and J.-P. Seifert, editors, *5th Workshop on Fault Diagnosis and Tolerance in Cryptography : FDTC 2008*, pages 92–98, Washington DC, United States, Aug. 2008. IEEE Computer Society Press.
4. S. D. Galbraith. *Mathematics of public key cryptography.* Cambridge University Press., 2012.
5. H. Krawczyk. [TLS] OPTLS: Signature-less TLS 1.3. November 1, 2014. *https://www.ietf.org/mail-archive/web/tls/current/msg14385.html.*
6. Jean-Pierre Flori, Jérôme Plût, Jean-René Reinhard. Diversity and transparency for ECC. *NIST workshop on ECC Standards, June 11-12, 2015.*
7. S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
8. T. Pornin. RFC 6797 Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). 2013.
9. V.G. Martínez, F.H. Álvarez, L.K. Encinas, C.S. Ávila:. A Comparison of the Standardized Versions of ECIES. *Sixth International Conference on Information Assurance and Security*, 2010.
10. W. Schindler, A. Wiemers. Power Attacks in the Presence of Exponent Blinding. *J Cryptogr Eng 4 (2014), 213-236.*
11. Werner Schindler and Andreas Wiemers. Efficient Side-Channel Attacks on Scalar Blinding on Elliptic Curves with Special Structure. *NIST Workshop on ECC Standards.*