

A Fault Attack on Pairing Based Cryptography

Dan Page¹ and Frederik Vercauteren²

Abstract—Current fault attacks against public key cryptography focus on traditional schemes such as RSA and ECC, and to a lesser extent primitives such as XTR. However, bilinear maps, or pairings, have presented theorists with a new and increasingly popular way of constructing cryptographic protocols. Most notably, this has resulted in efficient methods for Identity Based Encryption (IBE). Since identity based cryptography seems an ideal partner for identity aware devices such as smart-cards, in this paper we examine the security of concrete pairing instantiations in terms of fault attack.^{3 4}

Index Terms—Cryptography, fault attack, Tate pairing, identity based encryption

I. INTRODUCTION

The increasing ubiquity of computing devices is continuing to offer exciting new applications to consumers, but also multiplies the number of security issues a system designer must consider. Since such devices will be carried into and used in hostile environments and often house sensitive information, for example identity related tokens or financial information, the threat of attack is significant. This threat is magnified by both the potential pay-off and level of anonymity that side-channel attacks allow.

Side-channel analysis, formally proposed by Kocher et al. [12], [13], is the art of externally monitoring a device while it executes some algorithm that includes secret information. We can extend this passive definition of side-channel attack to include active, fault injection attacks [1], [2] whereby an attacker can physically manipulate the target device, tamper with internal state and perform erroneous operations. Although attack and defence techniques in this context are increasingly well understood, the current emphasis in terms of public key systems is mainly on traditional cryptographic schemes such as RSA and ECC. For example, in ECC based systems, one generally derives security from a discrete logarithm problem posed over the curve group. That is, one constructs some

private value d and performs the operation $Q = d \cdot P$ for a public point P . For large enough curve groups, reversal of this operation is intractable and hence one can transmit Q without revealing d . A multitude of side-channel and fault attacks against implementations of this primitive have been proposed. An equally large range of innovative countermeasures mean that when correctly implemented, the threats can generally be at least managed and possibly nullified.

There has been no previous work on the security of pairing based cryptography [9] in the context of side-channel and fault attack. Since pairings, formally described as bilinear maps, underpin cryptographic protocols such as Identity Based Encryption (IBE) [5], one might see them as an ideal application for the same identity aware, ubiquitous computing devices that are vulnerable to attacks of this type. One reason for this lack of analysis is the fact that early implementation techniques for computing the Tate pairing such as the BKLS algorithm [4] are effectively realised as a point multiplication with a fixed multiplier and some auxiliary operations to compute the pairing value. From such a description, the fact that the algorithm performs a fixed sequence of operations and has no secret in a conventional sense has led people to believe that the pairing is secure against current side-channel and fault attack methods. In part, this is true. However, the role of the pairing in associated protocols is somewhat different and much more diverse than point multiplication in ECC: it isn't enough to consider the new primitive in the context of traditional attack methods.

In an attempt to bridge the resulting gap, in this paper we present an investigation of pairing evaluation algorithms that are based on a closed algebraic expression, in terms of their security against fault attack. We organise our work by first presenting a brief introduction to pairings and algorithms for their evaluation in Section II. We then investigate applications of active, fault injection attacks against the Duursma-Lee algorithm and its extensions by Kwon and Barreto et al. in Section III. In Section IV we present efficient countermeasures and finally, we end with some concluding remarks in Section V.

II. AN INTRODUCTION TO PAIRINGS

Let E be an elliptic curve over a finite field \mathbb{F}_q , and let \mathcal{O} denote the identity element of the associated group of rational points $E(\mathbb{F}_q)$. For a positive integer $l \nmid \#E(\mathbb{F}_q)$ coprime to q , let \mathbb{F}_{q^k} be the smallest extension field of \mathbb{F}_q which contains the l -th roots of unity in $\overline{\mathbb{F}_q}$. The extension degree k is called the security multiplier. Let $E(\mathbb{F}_q)[l]$ denote the subgroup of $E(\mathbb{F}_q)$ of all points of order dividing l , and similarly for the degree k extension of \mathbb{F}_q . To unify the notation used in most protocols,

¹ Department of Computer Science, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol, BS8 1UB, United Kingdom. E-mail: page@cs.bris.ac.uk

² Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen) at Department of Electrical Engineering, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium. E-mail: fvercaut@esat.kuleuven.be

³The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability

⁴2006 IEEE. personal use of this material is permitted. however, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the ieee.

we use three groups \mathbf{G}_1 , \mathbf{G}_2 and \mathbf{G}_3 . The groups \mathbf{G}_1 and \mathbf{G}_2 will always be subgroups of elliptic curve groups, whereas the group \mathbf{G}_3 is a subgroup of the multiplicative group of a finite field.

A. The Reduced Tate Pairing

For a thorough treatment of the following, we refer the reader to [4] and also [10], and to [16] for an introduction to divisors. The Tate pairing of order l is the map

$$e_l : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_{q^k})[l] \rightarrow \mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^l,$$

given by $e_l(P, Q) = f_{P,l}(\mathcal{D})$. Here $f_{P,l}$ is a function on E whose divisor is equivalent to $l(P) - l(\mathcal{O})$, \mathcal{D} is a divisor equivalent to $(Q) - (\mathcal{O})$, whose support is disjoint from the support of $f_{P,l}$, and $f_{P,l}(\mathcal{D}) = \prod_i f_{P,l}(P_i)^{a_i}$, where $\mathcal{D} = \sum_i a_i P_i$. It satisfies the following properties:

- For each $P \neq \mathcal{O}$ there exists $Q \in E(\mathbb{F}_{q^k})[l]$ such that $e_l(P, Q) \neq 1 \in \mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^l$ (*non-degeneracy*).
- For any integer n , $e_l([n]P, Q) = e_l(P, [n]Q) = e_l(P, Q)^n$ for all $P \in E(\mathbb{F}_q)[l]$ and $Q \in E(\mathbb{F}_{q^k})[l]$ (*bilinearity*).
- Let $L = hl$. Then $e_l(P, Q)^{(q^k-1)/l} = e_L(P, Q)^{(q^k-1)/L}$.
- It is efficiently computable.

The non-degeneracy condition requires that Q is not a multiple of P , i.e. that Q is in some order l subgroup of $E(\mathbb{F}_{q^k})$ disjoint from $E(\mathbb{F}_q)[l]$. When one computes $f_{P,l}(\mathcal{D})$, the value obtained belongs to the quotient group $\mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^l$, and not $\mathbb{F}_{q^k}^*$. In this quotient, for a and b in $\mathbb{F}_{q^k}^*$, $a \sim b$ if and only if there exists $c \in \mathbb{F}_{q^k}^*$ such that $a = bc^l$. Clearly, this is equivalent to

$$a \sim b \text{ if and only if } a^{(q^k-1)/l} = b^{(q^k-1)/l},$$

and hence one ordinarily uses this value as the canonical representative of each coset. The isomorphism between $\mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^l$ and the elements of order l in $\mathbb{F}_{q^k}^*$ given by this exponentiation makes it possible to compute $f_{P,l}(Q)$ rather than $f_{P,l}(\mathcal{D})$. In fact we can consider the reduced Tate pairing as a pairing of the groups $\mathbf{G}_1 \times \mathbf{G}_2 \rightarrow \mathbf{G}_3$, with $\mathbf{G}_1 = \mathbf{G}_2 = E(\mathbb{F}_{q^k})[l]$ and $\mathbf{G}_3 = \mu_l \subset \mathbb{F}_{q^k}$, the group of l -th roots of unity.

B. The Tate Pairing on Supersingular Curves

From its very definition is clear that the Tate pairing can only be computed efficiently if the security multiplier k is small. Since by definition, k is the order of q in \mathbb{F}_l^* , it will for a general curve be the size of l . Therefore, the first curves to be used in pairing based cryptography were supersingular curves, since their security multiplier satisfies $k \leq 6$. The maximum $k = 6$ is only attained in characteristic 3, whereas for fields of characteristic 2 we have $k \leq 4$ and $k \leq 2$ otherwise. More recently, Miyaji, Nakabayashi and Takano [15] (MNT) devised a method to construct ordinary elliptic curves over large prime fields with security multiplier $k \leq 6$. The two main differences between ordinary and supersingular curves from a pairing perspective are as follows:

- Supersingular curves admit a distortion map, i.e. a non-rational endomorphism $\phi : E(\mathbb{F}_q) \rightarrow E(\mathbb{F}_{q^k})$. The

Algorithm 1: The Duursma-Lee algorithm.

Input : point $P = (x_1, y_1)$, point $Q = (x_2, y_2)$
Output: $f_P(\phi(Q)) \in \mu_l \subset \mathbb{F}_{q^6}^*$

```

1  $f \leftarrow 1$ 
2 for  $i = 1$  to  $m$  do
3    $x_1 \leftarrow x_1^3, y_1 \leftarrow y_1^3$ 
4    $\mu \leftarrow x_1 + x_2 + b$ 
5    $\lambda \leftarrow -y_1 y_2 \sigma - \mu^2$ 
6    $g \leftarrow \lambda - \mu \rho - \rho^2$ 
7    $f \leftarrow f \cdot g$ 
8    $x_2 \leftarrow x_2^{1/3}, y_2 \leftarrow y_2^{1/3}$ 
9 return  $f^{q^3-1}$ 

```

advantage of this distortion map is that it allows to define a non-trivial pairing on $E(\mathbb{F}_q)[l]$: let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be elements of $E(\mathbb{F}_q)[l]$, then the modified Tate pairing on $E(\mathbb{F}_q)[l]$ is the mapping $f_P(\phi(Q))$. Note that the image $\phi(Q) \notin E(\mathbb{F}_q)[l]$ and thus is linearly independent of P .

- There are known closed algebraic expressions for the modified Tate pairing on supersingular curves in characteristic 2 and 3. This not only enables very efficient implementations, but also makes the algorithms more vulnerable to fault attacks as we will show.

In the remainder of this paper, we will only consider supersingular curves in characteristic 2 and 3.

For \mathbb{F}_q with $q = 3^m$, the supersingular curves used in cryptography are defined by an equation of the form $E : Y^2 = X^3 - X + b$, with $b = \pm 1$. Let $\mathbb{F}_{q^3} = \mathbb{F}_q[\rho]/(\rho^3 - \rho - b)$, with $b = \pm 1$ depending on the curve equation, and let $\mathbb{F}_{q^6} = \mathbb{F}_{q^3}[\sigma]/(\sigma^2 + 1)$. Then there exists a distortion map $\phi : E(\mathbb{F}_q) \rightarrow E(\mathbb{F}_{q^6})$ defined by $\phi(x, y) = (\rho - x, \sigma y)$.

Duursma and Lee introduced a specialised algorithm [7] to compute pairings on a family of hyperelliptic curves, including the aforementioned supersingular curves in characteristic 3. The performance of their method was improved upon slightly by Kwon [8], by exchanging 2 cube roots for 8 cube operations and by Barreto, Galbraith, O'hEigeartaigh and Scott [3] who introduced the η and η_T pairings. The Duursma-Lee method is shown in Algorithm 1; note that the final result is powered by $q^3 - 1$ to form a compatible result with the BKLS algorithm [4].

In the case of $q = 2^m$ with m odd, the supersingular curves with $k = 4$ are given by equations of the form $E : y^2 + y = x^3 + x + b$ with $b \in \mathbb{F}_2$. Let $\mathbb{F}_{q^2} = \mathbb{F}_q[s]/(s^2 + s + 1)$ and $\mathbb{F}_{q^4} = \mathbb{F}_{q^2}[t]/(t^2 + t + s)$, then the distortion map $\phi : E(\mathbb{F}_q) \rightarrow E(\mathbb{F}_{q^4})$ is defined by $\phi(x, y) = (x + s^2, y + sx + t)$. Note that $s = t^5$ and that t satisfies $t^4 = t + 1$, so we can also represent \mathbb{F}_{q^4} as $\mathbb{F}_q[t]/(t^4 + t + 1)$. For points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ in $E(\mathbb{F}_q)[l]$, the modified Tate pairing is defined again as $f_P(\phi(Q))$.

The Duursma-Lee algorithm was independently extended to characteristic 2 by Kwon [8] and improved substantially by Barreto, Galbraith, O'hEigeartaigh and Scott [3] using the η and η_T approach. A careful comparison of the η pairing with Kwon's algorithm shows that they are in fact identical up to a slight rearrangement of terms; the Kwon-BGOS method is

Algorithm 2: The Kwon-BGOS algorithm.

Input : point $P = (x_1, y_1)$, point $Q = (x_2, y_2)$
Output: $f_P(\phi(Q)) \in \mu_l \subset \mathbb{F}_{q^4}^*$

```

1  $f \leftarrow 1$ 
2 for  $i = 1$  to  $m$  do
3    $x_1 \leftarrow x_1^2, y_1 \leftarrow y_1^2$ 
4    $\mu \leftarrow x_1 + x_2$ 
5    $\lambda \leftarrow \mu + x_1 x_2 + y_1 + y_2 + b$ 
6    $g \leftarrow \lambda + \mu t + (\mu + 1)t^2$ 
7    $f \leftarrow f \cdot g$ 
8    $x_2 \leftarrow \sqrt{x_2}, y_2 \leftarrow \sqrt{y_2}$ 
9 return  $f^{q^2-1}$ 

```

given in Algorithm 2. Note that the final result is powered by $q^2 - 1$ to be compatible with the BKLS algorithm. Finally, we mention that the fault attack described in Section III also applies with minor modifications to the η_T pairing and that an alternative implementation of the above algorithms might precompute the cube roots and square roots, which could possibly lead to other interesting vulnerabilities.

III. ATTACKING THE PAIRING

A. A Faulty Duursma-Lee Algorithm

1) *Recovering a Secret Point:* Given the secret point $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, a point chosen by the attacker, assume for the moment that we can eliminate the final powering. Let \bar{e}_Δ denote pairing via the Duursma-Lee algorithm where through tampering we induce some transient fault and replace the loop bound m with Δ . This could be achieved by using glitch attack [1], [14] to tamper with the loop test and branch mechanism, or given that the Duursma-Lee algorithm is essentially parameterised by m given the right field arithmetic and that m is therefore likely to be held in memory somewhere, by selectively provoking errors in memory or registers [2], [17]. Given this ability, instead of producing a product of polynomials of the form

$$\prod_{i=1}^m \left[-y_1^{3^i} \cdot y_2^{3^{m-i+1}} \sigma - \mu_i^2 - \mu_i \rho - \rho^2 \right]$$

with $\mu_i = x_1^{3^i} + x_2^{3^{m-i+1}} + b$, the algorithm instead produces

$$\prod_{i=1}^{\Delta} \left[-y_1^{3^i} \cdot y_2^{3^{m-i+1}} \sigma - \mu_i^2 - \mu_i \rho - \rho^2 \right]$$

for some value Δ and remembering that for now, we ignore the final powering.

If we were able to induce a fault so that $\Delta = m + 1$, for example by glitching the loop iteration so it executes once too often, then recovering the secret point is trivial: we compute two pairings, one correct and one tampered with

$$\begin{aligned} R_1 &= \bar{e}_m(P, Q) \\ R_2 &= \bar{e}_{m+1}(P, Q). \end{aligned}$$

If we use $g_{(i)}$ to denote the i -th factor of a product produced by the Duursma-Lee algorithm, by dividing the two results

above we are left with a single factor

$$g_{(m+1)} = -y_1^{3^{m+1}} \cdot y_2 \sigma - \mu_{m+1}^2 - \mu_{m+1} \rho - \rho^2.$$

Since we have that $z^{3^m} = z$ for all elements $z \in \mathbb{F}_q$, we can extract x_1 or y_1 , given that we know x_2 and y_2 , and hence reconstruct the secret point.

However, the ability to selectively induce a fault so $\Delta = m + 1$ seems tenuous. It is far more realistic to assume that we can induce $\Delta = m \pm r$ for random, unknown values of r . Using this ability, we calculate many erroneous pairing values with the aim of collecting a pair

$$\begin{aligned} R_1 &= \bar{e}_{m \pm r}(P, Q) \\ R_2 &= \bar{e}_{m \pm r + 1}(P, Q), \end{aligned}$$

so that we are again left with a single term of the product $g_{(m \pm r + 1)}$ and can hence run a similar method as above although needing to compensate for the differing powers of x_1, y_1, x_2 and y_2 introduced by r .

The problem with this approach is detecting when we have induced faults with the correct Δ values for use. This is also quite an easy task. Since the algorithm is straight-line and takes a fixed number of operations, given the time taken to compute the pairing or a power profile, it seems simple to work out how many loop iterations were performed in the same way that one can observe and count the sixteen rounds of DES. Given the value of r , this approach will deterministically recover the correct value of P and requires reasonably few invocations of the pairing on the target device due to a similar argument as the birthday paradox: we simply keep provoking random faults until we recover a pair of results that satisfy our conditions.

2) *Reversing the Final Powering:* The remaining problem then, is to reverse the final powering which takes the output of the pairing function and produces a unique representative for the coset in $\mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^l$. To reverse this operation given the result $R = e(P, Q)$ we want to recover S , the value that was computed by the algorithm before the final powering so that $R = S^{(q^k - 1)/l}$. For the Duursma-Lee method shown in Algorithm 1, this simplifies to $R = S^{q^3 - 1}$. It is clear that given R , the value of S is not uniquely determined, but only up to a non-zero factor in \mathbb{F}_{q^3} . Indeed, for non-zero $c \in \mathbb{F}_{q^3}$ we have $c^{q^3 - 1} = 1$. Furthermore, given one solution $S \in \mathbb{F}_{q^6}$ to $X^{q^3 - 1} - R = 0$, all others are of the form $c \cdot S$ for non-zero $c \in \mathbb{F}_{q^3}$.

However, given the attack description above we are not trying to reverse the powering on the full product of factors computed by the Duursma-Lee algorithm, rather only one of these factors which has a fairly special form. That is, given

$$R = \frac{R_2}{R_1} = \frac{\bar{e}_{m \pm r + 1}(P, Q)}{\bar{e}_{m \pm r}(P, Q)} = g_{(m \pm r + 1)}^{q^3 - 1}$$

we want to recover $g_{(m \pm r + 1)}$ which will, in turn, allow us to recover the correct x_1 and y_1 for the secret point.

We therefore need a method to compute one valid root of $R = g^{q^3 - 1}$ for some factor g , and then derive the correct value of g from among all possible solutions. The first problem can

be solved very efficiently as follows: multiply the equation $X^{q^3-1} - R = 0$ by X to obtain

$$X^q - R \cdot X = 0,$$

and note that the operator $X^q - R \cdot X$ is a linear operator on the two dimensional vector space $\mathbb{F}_{q^6}/\mathbb{F}_{q^3}$. Since $\mathbb{F}_{q^6} \cong \mathbb{F}_{q^3}[\sigma]/(\sigma^2-1)$ we can write $X = x_0 + \sigma x_1$ and $R = r_0 + \sigma r_1$, with $x_0, x_1, r_0, r_1 \in \mathbb{F}_{q^3}$. Using this representation, we see that the above equation is equivalent to

$$M \cdot X = \begin{pmatrix} 1-r_0 & r_1 \\ r_1 & 1+r_0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = 0.$$

The kernel of the matrix M is a one-dimensional vector space over \mathbb{F}_{q^3} and thus provides all the solutions to $X^{q^3-1} - R = 0$.

To choose the correct root among all q^3-1 possibilities, we use the specific form of the factors in the product computed in the Duursma-Lee algorithm. Indeed, each factor g is of the form

$$g = g_0 + g_1\rho - \rho^2 + g_2\sigma,$$

with $g_0, g_1, g_2 \in \mathbb{F}_q$. To recover g from $R = g^{q^3-1}$, we first obtain $g' = c \cdot g$ for some $c \in \mathbb{F}_{q^3}$ using the root finding algorithm above, and then compute c^{-1} and thus g itself. Again this boils down to a simple linear system of equations: by multiplying g' with an appropriate factor in \mathbb{F}_{q^3} , we can assume that g' is of the form $g' = 1 + (g'_0 + g'_1\rho + g'_2\rho^2)\sigma$. Let $d = c^{-1} = g/g' \in \mathbb{F}_{q^3}$, then d clearly is of the form $d = d_0 + d_1\rho - \rho^2$. To determine $d_0, d_1 \in \mathbb{F}_q$, we use the fact that the terms $\rho\sigma$ and $\rho^2\sigma$ do not appear in g . This finally gives the following linear system of equations:

$$\begin{pmatrix} g'_1 & g'_0 + g'_2 \\ g'_2 & g'_1 \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \end{pmatrix} = \begin{pmatrix} g'_1 + g'_2 \\ g'_0 + g'_2 \end{pmatrix}.$$

B. A Faulty Kwon-BGOS Algorithm

1) *Recovering a Secret Point*: The attack follows closely the attack described for the Duursma-Lee algorithm; the main difference being the exact form of the factors computed in Algorithm 2. Again assume we want to recover a secret point $P = (x_1, y_1)$ and have knowledge of the point $Q = (x_2, y_2)$. By changing the loop bound to Δ , the Kwon-BGOS algorithm now computes the following product

$$\prod_{i=1}^{\Delta} [\mu_i + \tau_i + b + \mu_i t + (\mu_i + 1)t^2],$$

with $\mu_i = x_1^{2^i} + x_2^{2^{m-i+1}}$ and $\tau_i = x_1^{2^i} x_2^{2^{m-i+1}} + y_1^{2^i} + y_2^{2^{m-i+1}}$. It is clear that if we are given one factor of the above product and the point Q , we can recover the secret point P , since the coefficient of t gives x_1 and the constant coefficient gives y_1 .

2) *Reversing the Final Powering*: As for the Duursma-Lee algorithm, we can recover P if we are given the correct factor of the product; however, by running the faulty Kwon-BGOS algorithm, we actually get

$$R = g_{(m \pm r + 1)}^{q^2-1},$$

which only determines $g_{(m \pm r + 1)}$ up to a factor in $\mathbb{F}_{q^2}^*$. Again we proceed in two steps: first we recover a random (q^2-1) -th

root of R , and then exploit the structure of the factor $g_{(m \pm r + 1)}$ to compensate for the random factor in $\mathbb{F}_{q^2}^*$.

To solve $X^{q^2-1} - R = 0$, we consider the linear operator $X^{q^2} - R \cdot X$ on the 2-dimensional vector space $\mathbb{F}_{q^4}/\mathbb{F}_{q^2}$. Recall that $\mathbb{F}_{q^4} = \mathbb{F}_{q^2}[t]/(t^2 + t + s)$ and $\mathbb{F}_{q^2} = \mathbb{F}_q[s]/(s^2 + s + 1)$. Let $X = x_0 + x_1 t$ and $R = r_0 + r_1 t$ with $x_0, x_1, r_0, r_1 \in \mathbb{F}_{q^2}$, then the above equation is equivalent to

$$M \cdot X = \begin{pmatrix} 1+r_0 & 1+r_1 s \\ r_1 & 1+r_0+r_1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = 0.$$

Any non-zero solution (g_0, g_1) gives a valid (q^2-1) -th root $g = g_0 + g_1 t$ of R . Furthermore, any other root is given by cg with $c \in \mathbb{F}_{q^2}^*$.

To find the correct root, we exploit the structure of $g_{(m \pm r + 1)}$; taking into account that $t^2 = t + s$, $g_{(m \pm r + 1)}$ is given by an expression of the form

$$g_k = \mu_k + \tau_k + b + (\mu_k + 1)s + t,$$

for $k = m \pm r + 1$. In particular, there is no st -term, and the coefficient of t is 1. Assume we have computed the root $g = g_0 + g_1 t$, then the correct root is clearly given by $g_1^{-1} g = g_0 g_1^{-1} + t$.

IV. COUNTERMEASURES

In this section we investigate several types of countermeasures specifically tailored to pairing based cryptography.

A. Using New Point Blinding Techniques

As shown in the previous section, the attack on the pairing algorithms is only successful when the adversary has knowledge of one input point. Therefore it is natural to utilise point blinding techniques to construct a defence mechanism. Such techniques apply a blinding factor to one or both of the input points before the pairing and eliminate this factor afterwards. If we blind the point under control of the attacker, any tampering with the pairing will cause the un-blinding phase to essentially produce a random result. Since we know that the relationship

$$e(a \cdot P, b \cdot Q) = e(P, Q)^{a \cdot b}$$

holds, we can randomise the points P and Q by selecting random values for a and b . Although this results in an additional factor of $a \cdot b$ in the exponent of the result, we can eliminate this by careful selection of a and b such that $a \cdot b = 1 \pmod{l}$. Note that for the modified Tate pairing on supersingular curves, both P and Q are defined over \mathbb{F}_q , so computing $b \cdot Q$ has the same cost as computing $a \cdot P$.

If one cannot accept the cost of a GCD-like operation to compute arbitrary random a and b pairs of the right form, it is possible to employ a deterministic update procedure akin to traditional point blinding. One computes and stores two random pairs such that

$$\begin{aligned} a \cdot b &= 1 \pmod{l} \\ c \cdot d &= 1 \pmod{l}. \end{aligned}$$

This is only ever done once, perhaps when the device is initialised. Updating a and b can then be achieved by multiplication with c and d

$$\begin{aligned} a &\leftarrow a \cdot c \pmod{l} \\ b &\leftarrow b \cdot d \pmod{l} \end{aligned}$$

which only costs two field multiplications per update and retains the fact that $a \cdot b = 1 \pmod{l}$. In summary, the defence costs two additional point multiplications and two field multiplications to instrument.

B. Altering Traditional Point Blinding

Traditional point blinding for ECC has the host device store two extra points that are retained across invocations of the point multiplication algorithm. Given a secret multiplier d , the card stores a random point R and the point $S = d \cdot R$. The point multiplication $d \cdot P$ is then computed as $d \cdot P = d \cdot (P + R) - S$ so that the point fed into the multiplication routine is randomly blinded by first adding R with the result recovered by subtracting S . Considering the relationship

$$e(P, Q + R) = e(P, Q) \cdot e(P, R)$$

we can apply an augmented point blinding technique to the pairing so that the controlled point is again randomised before use. For the DPA-like attack described above, this seems sufficient defence since by randomising the control point, the attack can no longer correctly compute the multiplication oracle that drives DPA selection.

Assuming P is the secret point, Q is the point controlled by the attacker that we want to randomise, R is a random point and $S = e(P, R)^{-1}$, we apply the map as

$$e(P, Q) = e(P, Q + R) \cdot S.$$

By blinding Q , the point under control of the attacker, we effectively remove this control and hence defeat any attack based on it: the attacker can no longer reason about internal operation based on the point they sent into the pairing.

Since P and R are known to the device at initialisation, we can store the point R and field element S in order to implement the method. Using this technique, the overhead in performance is equal to one point addition and a field multiplication in \mathbf{G}_3 . However, the issue of updating the blinding variables presents a drawback. Updating R and S is performed before or after the multiplication routine is executed in order to provide changing and hopefully non-deterministic blinding factors to the real point. In the traditional blinding defence, one might select a random $b \in \{-2, +2\}$ and then set $R \leftarrow b \cdot R$ and $S \leftarrow b \cdot S$.

In our case, we need to replace the update of S with $S \leftarrow S^b$. Squaring a field element is reasonably inexpensive but in order to accommodate both potential values of b , we need to store some extra information so that inversion is equally inexpensive. To do this, we can simply store and update the value S^{-1} along with S if there is enough space or use specialised representations of \mathbf{G}_3 which renders inversion inexpensive such as [11] in characteristic three.

V. CONCLUSION

We have presented the first investigation into the security of pairing based cryptography against side-channel attack. Although the use of pairings in the sorts of environment where side-channel attack is most prevalent is still some way off, the coupling of identity based cryptography with identity aware devices seems very attractive. Naive early assumptions about the security of pairings in this setting were as a result of directly applying existing knowledge of RSA and ECC vulnerabilities. However, since the use of pairings represents a vastly different and more diverse problem from that in more conventional systems, it is important to also consider new methods of attack. Our results show that although vulnerabilities do inevitably exist, creative use of the bilinearity property of pairings and sensible implementation methods help to minimise such risk with low overhead.

Since the topic of pairings is fairly new, there is plenty of further work that could be done on fault attacks. Ciet and Joye [6] described two different fault attacks on ECC which work by computing point multiplications with invalid points and erroneous field arithmetic. Clearly, these attacks still apply whenever point multiplications are computed in pairing based systems. Generally, it seems an open problem to consider what happens in the context of pairings if the curve parameters are erroneous; the input points are not on the correct curve; or the field representation is fault. These concepts seem interesting but more difficult than the ECC case since, for example, feeding invalid points into the pairing generally causes it to become degenerate.

It also seems vital to address the security of the BKLS [4] method of pairing evaluation. We have only considered closed-form style algorithms here; BKLS relies on a more general approach which is much more similar to ECC point multiplication. Since the BKLS algorithm seems an attractive choice for implementing systems based on fields of large prime characteristic, perhaps using MNT curves [15], addressing security against fault attack is an interesting open problem.

ACKNOWLEDGEMENTS

The authors would like to thank Steven Galbraith, Rob Granger, Nigel Smart and Martijn Stam for invaluable help and discussion throughout the course of this work.

REFERENCES

- [1] R.J. Anderson and M.G. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In *International Security Protocols Workshop (IWSP)*, Springer-Verlag LNCS 1361, 125–136, 1997.
- [2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall and C. Whelan. The Sorcerer's Apprentice Guide to Fault Attacks, In *Cryptology ePrint Archive*, Report 2004/10, 2004.
- [3] P.S.L.M. Barreto, S. Galbraith, C. O'hEigeartaigh and M. Scott. Efficient Pairing Computation on Supersingular Abelian Varieties. In *Cryptology ePrint Archive*, Report 2004/375, 2004.
- [4] P.S.L.M. Barreto, H. Kim, B. Lynn and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. In *Advances in Cryptology (CRYPTO)*, Springer-Verlag LNCS 2442, 354–368, 2002.
- [5] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *SIAM Journal on Computing*, Volume 32, no. 3, 586–615, 2003.

- [6] M. Ciet and M. Joye. Elliptic Curve Cryptosystems in the Presence of Permanent and Transient Faults. To appear in *Designs, Codes and Cryptography*, 2004.
- [7] I. Duursma and H. Lee. Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$. In *Advances in Cryptology (ASIACRYPT)*, Springer-Verlag LNCS 2894, 111–123, 2003.
- [8] S. Kwon. Efficient Tate Pairing Computation for Supersingular Elliptic Curves over Binary Fields. In *Cryptology ePrint Archive*, Report 2004/303, 2004.
- [9] R. Dutta, R. Barua and P. Sarkar, Pairing-Based Cryptographic Protocols : A Survey. In *Cryptology ePrint Archive*, Report 2004/064, 2004.
- [10] S. Galbraith, K. Harrison and D. Soldera. Implementing the Tate pairing. In *Algorithmic Number Theory Symposium (ANTS-V)*, Springer LNCS 2369, 324–337, 2002.
- [11] R. Granger, D. Page and M. Stam. On Small Characteristic Algebraic Tori in Pairing-Based Cryptography. In *Cryptology ePrint Archive*, Report 2004/132, 2004.
- [12] P.C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology (CRYPTO)*, Springer-Verlag LNCS 1109, 104–113, 1996.
- [13] P.C. Kocher, J. Jaffe and B. Jun. Differential Power Analysis. In *Advances in Cryptology (CRYPTO)*, Springer-Verlag LNCS 2139, 388–397, 1999.
- [14] O. Kömmerling and M.G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *USENIX Workshop on Smart Card Technology*, 9–20, 1999.
- [15] A. Miyaji, M. Nakabayashi and S. Takano. New Explicit Conditions on Elliptic Curve Traces for FR-reduction. In *IEICE Trans. Fundamentals* E-84 A(5), 1234–1243, 2001.
- [16] J. Silverman. The Arithmetic of Elliptic Curves. Springer-Verlag GTM 106, 1986.
- [17] S.P. Skorobogatov and R.J. Anderson. Optical Fault Induction Attacks. In *Cryptographic Hardware and Embedded Systems (CHES)*, Springer-Verlag LNCS 2523, 2–12, 2002.



Daniel Page is a BSc graduate of the University of Nottingham and received a PhD from the University of Bristol where he conducted research into cache systems. Having published around 20 research papers, he has recently taken up a lectureship at the University of Bristol in the interdisciplinary area of computer architecture and cryptography.



Frederik Vercauteren received the MSc degree in computer science, the MSc degree in pure mathematics and the PhD degree in electrical engineering from the Katholieke Universiteit Leuven, Belgium. He is currently a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen) at the Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium. Previously, he was a lecturer at the Department of Computer Science, University of Bristol, United Kingdom. His current research interests include applications of computational number theory and arithmetic geometry in cryptography.