Optimal generic attack against basic Boneh-Boyen signatures

Yen-Kang Fu[1], Jonathan Chang[1], and David Jao[1,2]

[1] Department of Combinatorics and Optimization, University of Waterloo
200 University Ave. W, Waterloo, Ontario N2L 3G1, Canada
{ted.fu,jonathan.chang,djao}@uwaterloo.ca
[2] evolutionQ Inc., 2B-180 Columbia St. W, Waterloo, Ontario N2L 3L3, Canada
david.jao@evolutionq.com

Abstract. The Boneh-Boyen digital signature scheme is a pairing-based signature scheme that features short signatures consisting of one group element, the minimum possible size. In contrast to earlier short signature schemes such as Boneh-Lynn-Shacham, the Boneh-Boyen scheme achieves security without the use of random oracles, but at the cost of a non-standard mathematical assumption, the q-Strong Diffie-Hellman (or q-SDH) assumption, which is known to be less secure than discrete logarithms against generic attacks. However, unlike discrete logarithms, in which the fastest known generic attacks match the known provable lower bounds for solving generic discrete logarithms, the fastest known generic attacks against Boneh-Boyen prior to this work did not match the provable lower bounds for generically solving q-SDH instances. In this work, we demonstrate that when $p-1$ has suitably sized divisors (where p is the order of the underlying group used in the scheme), which in particular almost always occurs for cryptographic pairings instantiated from elliptic curves, Boneh-Boyen can indeed be broken in the sense of weak existential forgery under chosen-message attack (the same security definition as what was used in the original Boneh-Boyen paper) in $O(p^{\frac{1}{3}+\varepsilon})$ time using generic algorithms, matching the provable lower bound for generically solving q-SDH instances.

1 Introduction

The era of pairing-based cryptography is in many ways defined by the publication of the Boneh-Franklin identity-based encryption (IBE) scheme [6,7] and the associated Boneh-Lynn-Shacham (BLS) digital signature scheme [8,9]. These two cryptosystems presented the first compelling evidence that pairing-based cryptography could provide useful functionality beyond anything achievable using traditional means — IBE in the case of Boneh-Franklin, and short signatures in the case of BLS. In both of these schemes, security is proved in the random oracle model (ROM), which entails making a heuristic assumption that a hash function behaves like a random function. As an alternative, in order to avoid the random oracle assumption, Boneh and Boyen subsequently published an

IBE scheme [2,5] and companion signature scheme [3,4] whose security could be proved in the standard model. However, the trade-off is that the Boneh-Boyen schemes require the use of a non-standard mathematical security assumption, the so-called q-SDH assumption (§2.3). The tension between these two alternative sets of trade-offs and assumptions is documented in the "Another Look" series of papers by Koblitz and Menezes [17,18].

One piece of information that helps to determine the correct trade-off is the exact level of security of the respective schemes and assumptions against an adversary. For this purpose, we need consider only the signature schemes, since the companion IBE schemes use the same assumptions; indeed, pairs of corresponding IBE and signature schemes are directly related via a generic transformation, as described in [6] in an observation attributed to Naor. The highlight of the BLS signature scheme is that a signature requires only a single group element, which is half the number of elements for a signature in a non-pairing based signature scheme such as Schnorr [20,21]. We remark that subsequent research has shown that solving discrete logarithms on pairing-friendly curves via transfer to a finite field is easier than previously thought [1], implying that the size advantage of BLS in practice is not as great as what it would otherwise be if only generic discrete log algorithms were employed [15], which somewhat blunts the size advantage of BLS. Nevertheless, in this work, for simplicity we only consider the security of BLS and Boneh-Boyen against generic attacks, that is, attacks which work in the generic bilinear group model of [2]. These results may be viewed as conclusions about the *mathematical* security of BLS and Boneh-Boyen in the setting where we use an idealized cryptographic pairing admitting no non-generic attacks, as opposed to the concrete setting of an actual elliptic curve-based pairing where the far messier transfer attacks come into play. This setting is the natural one to use when evaluating the amount of security degradation attributable specifically to the use of the non-standard q-SDH assumption.

Boneh-Boyen signatures come in two variants, a "basic" scheme which is only weakly unforgeable (§2.1) and a "full" scheme which is strongly unforgeable. If we ignore the fact that the basic Boneh-Boyen scheme provides only weak unforgeability, the basic scheme, at least on paper, matches the BLS scheme in size efficiency, in that a basic Boneh-Boyen signature consists of a single group element. However, we have known for some time that the q-SDH problem, on which Boneh-Boyen is based, is easier than the discrete logarithm problem in a generic group. Algorithms[3] by Brown and Gallant [10] and Cheon [11,12] provide up to a square-root speedup in solving q-SDH: that is, q-SDH can be solved generically in $O(p^{\frac{1}{4}})$ time, compared to $O(p^{\frac{1}{2}})$ time for discrete log. That said, achieving the $O(p^{\frac{1}{4}})$ figure requires using a specific value of q which is itself close to $p^{\frac{1}{2}}$. In the context of Boneh-Boyen, the value of q represents the number of signing queries, which cannot be close to $p^{\frac{1}{2}}$, since performing such a large number of signing queries would take more time than simply solving discrete log

[3] These algorithms are usually attributed to Cheon, even though Brown and Gallant were first, because Cheon was the first to apply these algorithms specifically to q-SDH.

from scratch. When taking into account the cost of signing queries, the fastest known generic attack against basic Boneh-Boyen is the known-message attack of Jao and Yoshida [16], which achieves a running time of $O(p^{\frac{2}{5}})$ in most circumstances; a similar chosen-message attack against full Boneh-Boyen is also given in [16], with the same running time. This attack outperforms generic discrete log (which takes $O(p^{\frac{1}{2}})$ time), showing definitively that Boneh-Boyen and q-SDH respectively do not achieve the same level of security as BLS and discrete log in the generic bilinear group model. However, until now, there still remained a gap between the $O(p^{\frac{2}{5}})$ attack against Boneh-Boyen from [16] and the known lower bound of $O(p^{\frac{1}{3}})$, proven in [4], for solving q-SDH in a generic group (for $q < p^{\frac{1}{3}}$).

1.1 Our contributions

In this work, we establish for the first time that the basic Boneh-Boyen scheme can be broken in $O(p^{\frac{1}{3}+\varepsilon})$ time in the generic bilinear group model, matching the $O(p^{\frac{1}{3}})$ lower bound proven in [4] and showing that this bound is achieved. That is, we present a concrete attack which breaks the weak unforgeability of basic Boneh-Boyen signatures in $O(p^{\frac{1}{3}+\varepsilon})$ time, whenever $p-1$ has a divisor of size approximately $p^{\frac{1}{3}}$. Compared to previous work [16], our result requires a divisor $d \mid p-1$ satisfying $d \approx p^{\frac{1}{3}}$, whereas [16] requires a divisor $d \mid p \pm 1$ satisfying $d \approx p^{\frac{2}{5}}$. Moreover, our result requires a chosen-message attack, compared to [16] which utilizes only a known-message attack. By choosing specific messages for our signing queries, we can exploit the algebraic structure of Boneh-Boyen signatures to speed up the reduction from weak forgery to q-SDH. Our attack uses Cheon's algorithm to solve q-SDH, and thus is considerably more powerful than a mere forgery, since Cheon's algorithm actually recovers the private key. However, Cheon's algorithm is not strictly necessary: our results also show that, in principle, any solution to the q-Cheon SDH problem (cf. §2.3) yields a basic Boneh-Boyen forgery in $O(q)$ time (improving upon the $O(q^2)$ reductions given in [4,16]), although in practice no speedups for q-SDH or q-Cheon-SDH other than Cheon's algorithm are presently known.

2 Preliminaries

2.1 Security definitions

We begin by reviewing two security definitions used in the proof of security for the Boneh-Boyen signature. The definition of strong existential unforgeability corresponds to the standard notion of security for digital signatures used in the literature. The full Boneh-Boyen signature scheme (§3.2) satisfies strong existential unforgeability under certain assumptions. Weak existential unforgeability is a diluted version of strong existential unforgeability in which the adversary is required to commit to the list of messages used for signature queries in advance of being given the public key. The security of the basic Boneh-Boyen scheme

(§3.1) is proved using weak existential unforgeability. It is important to emphasize that weak existential unforgeability is not our invention; the definition we use is identical to that of Boneh and Boyen's original paper [3,4].

Strong Existential Unforgeability. Strong existential unforgeability is defined via the following game between the challenger and an adversary $\mathcal{A}$.

1. The challenger generates a key pair (PK, SK) and gives PK to $\mathcal{A}$.
2. $\mathcal{A}$ adaptively makes up to q_S queries for signatures of messages $m_1, \ldots, m_{q_S}$ of its choice. The challenger responds to the queries with q_S valid signatures $\sigma_1, \ldots, \sigma_{q_S}$ of $m_1, \ldots, m_{q_S}$, respectively.
3. Eventually, $\mathcal{A}$ outputs a message-signature pair (m_*, σ_*), and wins the game if $(m_*, \sigma_*) \neq (m_i, \sigma_i)$ for all $i = 1, \ldots, q_S$ and Verify(m_*, σ_*, PK) = `true`

Weak Existential Unforgeability. Weak existential unforgeability is defined via the following game between the challenger and an adversary $\mathcal{A}$.

1. $\mathcal{A}$ makes up to q_S queries for signatures of messages $m_1, \ldots, m_{q_S}$ of its choice.
2. The challenger generates a key pair (PK, SK) and gives PK to $\mathcal{A}$. Then the challenger responds to the queries with q_S valid signatures $\sigma_1, \ldots, \sigma_{q_S}$ of $m_1, \ldots, m_{q_S}$, respectively.
3. Eventually, $\mathcal{A}$ outputs a message-signature pair (m_*, σ_*), and wins the game if $(m_*, \sigma_*) \neq (m_i, \sigma_i)$ for all $i = 1, \ldots, q_S$ and Verify(m_*, σ_*, PK) = `true`

The adversary $\mathcal{A}$'s advantage, denoted Adv Sig W($\mathcal{A}$), is defined as the probability that $\mathcal{A}$ wins the above game, where the probability is taken over random choices made by the adversary $\mathcal{A}$ and the challenger.

Our work centers entirely around the basic Boneh-Boyen scheme (§3.1), which only uses weak existential unforgeability. Hence we only use weak existential unforgeability in Definition 1.

Definition 1. *An adversary $\mathcal{A}$ can (t, q_S, ϵ)-weakly break a signature scheme if $\mathcal{A}$ runs in time at most t, makes at most q_S queries to the challenger, and* Adv Sig W($\mathcal{A}$) $\geq \epsilon$. *We say a signature scheme is (t, q_S, ϵ)-EUF-CMA (existentially unforgeable under a chosen-message attack) if there does not exist an adversary that (t, q_S, ϵ)-weakly breaks it.*

2.2 Bilinear pairings

The Boneh-Boyen signature scheme makes use of bilinear pairings. Let $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups with prime order p. The operations in $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are written multiplicatively. A function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is called a bilinear pairing if it satisfies the following conditions:

– **Bilinearity:** For any $u_1, u_2, u \in \mathbb{G}_1$ and $v_1, v_2, v \in \mathbb{G}_2$,

$$e(u_1 u_2, v) = e(u_1, v) \cdot e(u_2, v)$$

$$e(u, v_1 v_2) = e(u, v_1) \cdot e(u, v_2)$$

– **Non-degeneracy:** There exists $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$ such that $e(u, v) \neq 1$.

We assume the pairing function and the group operations are efficiently computable. The pair $(\mathbb{G}_1, \mathbb{G}_2)$ is called a bilinear group pair.

2.3 SDH problems

q-SDH problem. Let q be a publicly known positive integer. Boneh and Boyen [4] define the q-Strong Diffie-Hellman (q-SDH) problem on the bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ as follows:

> Given a $(q+3)$-tuple $(g_1, g_1^x, \ldots, g_1^{x^q}, g_2, g_2^x) \in \mathbb{G}_1^{q+1} \times \mathbb{G}_2^2$ as input, output $(c, g_1^{\frac{1}{x+c}})$ for some $c \in \mathbb{Z}_p$ such that $x + c \not\equiv 0 \pmod{p}$.

For our analysis, we need a variant of the q-SDH problem in which only three powers of g_1 are given. As shown in Theorem 3 in §5, this variant is still enough to yield an attack against the Boneh-Boyen signature scheme. We call this variant the q-Cheon-SDH problem:

> Given a 5-tuple $(g_1, g_1^x, g_1^{x^q}, g_2, g_2^x) \in \mathbb{G}_1^3 \times \mathbb{G}_2^2$ as input, output $(c, g_1^{\frac{1}{x+c}})$ for some $c \in \mathbb{Z}_p$ such that $x + c \not\equiv 0 \pmod{p}$.

The advantage Adv q-SDH$(\mathcal{A})$ of an adversary $\mathcal{A}$ in solving the q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as

$$\text{Adv } q\text{-SDH}(\mathcal{A}) = \Pr\left[\mathcal{A}(g_1, g_1^x, \ldots, g_1^{x^q}, g_2, g_2^x) = (c, g_1^{\frac{1}{x+c}})\right]$$

where the probability is taken over random choices of generators $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ and $x \in \mathbb{Z}_p^*$ and the random choices made by the adversary $\mathcal{A}$. Similarly, we define Adv q-Cheon-SDH$(\mathcal{A})$ as the advantage of an adversary $\mathcal{A}$ in solving the q-Cheon-SDH problem.

Definition 2. *An algorithm $\mathcal{A}$ can (t, ϵ)-break the q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ if $\mathcal{A}$ runs in time t and* Adv q-SDH$(\mathcal{A}) \geq \epsilon$. *We say that the (q, t, ϵ)-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if there is no algorithm that (t, ϵ)-breaks the q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$.*

Definition 3. *An algorithm $\mathcal{A}$ can (t, ϵ)-break the q-Cheon-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ if $\mathcal{A}$ runs in time t and* Adv q-Cheon-SDH$(\mathcal{A}) \geq \epsilon$. *We say that the (q, t, ϵ)-Cheon-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if there is no algorithm that (t, ϵ)-breaks the q-Cheon-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$.*

3 Boneh-Boyen signature scheme

Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of order p, and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear pairing. In [4], Boneh and Boyen present the following two versions of their signature schemes.

3.1 The Basic Signature Scheme

- **Key Generation:** KeyGen outputs random generators g_1 and g_2 of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and a random integer $x \in \mathbb{Z}_p^*$. Let $\zeta \leftarrow e(g_1, g_2) \in \mathbb{G}_T$. The public key is $\mathrm{PK} = (g_1, g_2, g_2^x, \zeta)$, and the secret key is $\mathrm{SK} = (g_1, x)$.
- **Signing:** Given a message $m \in \mathbb{Z}_p$ and a secret key SK, Sign(m, SK) outputs a signature $\sigma \leftarrow g_1^{\frac{1}{x+m}}$ where the exponent is calculated modulo p. In the event that $x + m \equiv 0 \pmod{p}$, Sign(m, SK) outputs $\sigma \leftarrow 1$.
- **Verification:** Verify(m, σ, PK) = `true` if and only if $e(\sigma, g_2^x \cdot g_2^m) = \zeta$.

3.2 The Full Signature Scheme

- **Key Generation:** KeyGen outputs random generators g_1 and g_2 of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and random integers $x, y \in \mathbb{Z}_p^*$. Let $\zeta \leftarrow e(g_1, g_2) \in \mathbb{G}_T$. The public key is $\mathrm{PK} = (g_1, g_2, g_2^x, g_2^y, \zeta)$, and the secret key is $\mathrm{SK} = (g_1, x, y)$.
- **Signing:** Given a message $m \in \mathbb{Z}_p$ and a secret key SK, Sign(m, SK) randomly picks $r \in \mathbb{Z}_p$ such that $x + m + yr \not\equiv 0$ and calculates $\sigma \leftarrow g_1^{\frac{1}{x+m+yr}}$. The signature is (σ, r).
- **Verification:** Verify(m, (σ, r), PK) = `true` if and only if $e(\sigma, g_2^x \cdot g_2^m \cdot (g_2^y)^r) = \zeta$.

In general, g_1 can be omitted from the public key with no loss of functionality.

4 Chosen message attack on the basic scheme

Proposition 1 is the main result of [16]. Using Proposition 1, the authors of [16] show that forging basic Boneh-Boyen signatures under a known-message attack, using q signing queries, reduces to solving q-SDH, via a reduction which costs $O(q^2)$ time. We show using Corollary 1 that Proposition 1 can be refined so as to yield a reduction to q-Cheon-SDH costing only $O(q)$ time, albeit under a chosen-message attack instead of a known-message attack.

Proposition 1. *Let $\mathbb{G}$ be a cyclic group of order p, let $g \in \mathbb{G}$ be a generator, and let $x \in \mathbb{Z}_p$. Let m_i for $i = 1, \ldots, d$ be distinct elements of $\mathbb{Z}_p$ such that $x + m_i \neq 0$. Then*

$$g^{\frac{x^k}{\prod_{i=1}^d (x+m_i)}} = \begin{cases} \displaystyle\prod_{i=1}^{d} g^{\frac{(-m_i)^k}{(x+m_i)\prod_{j\neq i}(m_j - m_i)}} & \text{for } 0 \leq k < d \\ \displaystyle g \cdot \prod_{i=1}^{d} g^{\frac{(-m_i)^d}{(x+m_i)\prod_{j\neq i}(m_j - m_i)}} & \text{for } k = d \\ \displaystyle g^x \cdot g^{-\sum_{i=1}^d m_i} \cdot \prod_{i=1}^{d} g^{\frac{(-m_i)^{d+1}}{(x+m_i)\prod_{j\neq i}(m_j - m_i)}} & \text{for } k = d+1 \end{cases}$$

Assume that all values m_i and $g^{\frac{1}{x+m_i}}$ are known. Furthermore, assume for $k = d$ and $k = d+1$ that g is known, and for $k = d+1$ that g^x is known. Then calculating

$g^{\frac{x^k}{\prod_{i=1}^{d}(x+m_i)}}$ *for a single* k *takes* $\Theta(dT+d^2T_p)$ *time, where* T *is the maximum time needed for a single exponentiation in* $\mathbb{G}$*, and* T_p *is the maximum time needed for an operation in* $\mathbb{Z}_p$*. Calculating all of* $g^{\frac{1}{\prod_{i=1}^{d}(x+m_i)}}, g^{\frac{x}{\prod_{i=1}^{d}(x+m_i)}}, \ldots, g^{\frac{x^{d+1}}{\prod_{i=1}^{d}(x+m_i)}}$ *takes* $\Theta(d^2T)$ *time.*

Proof. [16, Prop. 4.1].

Lemma 1. *Let* $\xi \in \mathbb{Z}_p$ *and* $m_i = \xi^i$ *for* $i = 1, 2, \ldots, d$*. Suppose the* m_i*'s are all distinct. For* $1 \leq i \leq d$*, define*

$$D_i = \prod_{\substack{j=1 \\ j \neq i}}^{d} (m_j - m_i)$$

Then, for all $0 < i < d$*,*

$$D_{i+1} = D_i \cdot \xi^d \cdot \frac{1-\xi^i}{\xi^{d+1}-\xi^{i+1}}$$

Proof. We write

$$\begin{aligned} D_i &= (m_1 - m_i) \cdots (m_{i-1} - m_i)(m_{i+1} - m_i) \cdots (m_d - m_i) \\ D_{i+1} &= (m_1 - m_{i+1}) \cdots (m_i - m_{i+1})(m_{i+2} - m_{i+1}) \cdots (m_d - m_{i+1}) \end{aligned}$$

Note that

$$\begin{aligned} (m_2 - m_{i+1}) &= (\xi^2 - \xi^{i+1}) = \xi(\xi - \xi^i) = \xi(m_1 - m_i) \\ (m_3 - m_{i+1}) &= (\xi^3 - \xi^{i+1}) = \xi(\xi^2 - \xi^i) = \xi(m_2 - m_i) \\ &\vdots \\ (m_i - m_{i+1}) &= (\xi^i - \xi^{i+1}) = \xi(\xi^{i-1} - \xi^i) = \xi(m_{i-1} - m_i) \\ (m_{i+2} - m_{i+1}) &= (\xi^{i+2} - \xi^{i+1}) = \xi(\xi^{i+1} - \xi^i) = \xi(m_{i+1} - m_i) \\ &\vdots \\ (m_d - m_{i+1}) &= (\xi^d - \xi^{i+1}) = \xi(\xi^{d-1} - \xi^i) = \xi(m_{d-1} - m_i) \end{aligned}$$

Hence every factor of D_{i+1} except for $(m_1 - m_{i+1})$ is equal to ξ times one of the factors of D_i. The expression D_{i+1} consists of $d-1$ factors, of which $d-2$ of the factors are equal to ξ times a factor of D_i. The only factor of D_{i+1} not paired up with a factor of D_i is $(m_1 - m_{i+1})$, and the only factor of D_i not paired up with a factor of D_{i+1} is $(m_d - m_i)$. Therefore the quotient D_{i+1}/D_i is equal to

$$\frac{D_{i+1}}{D_i} = \xi^{d-2}\frac{m_1 - m_{i+1}}{m_d - m_i} = \xi^{d-2}\frac{\xi - \xi^{i+1}}{\xi^d - \xi^i} = \xi^{d-2} \cdot \frac{\xi}{\xi} \cdot \frac{\xi - \xi^{i+1}}{\xi^d - \xi^i} = \xi^d \frac{1-\xi^i}{\xi^{d+1} - \xi^{i+1}}.$$

Corollary 1. *In Proposition 1, suppose that there exists $\xi \in \mathbb{Z}_p$ such that $m_i = \xi^i$ for $i = 1, 2, \ldots, d$, and the m_i's are distinct. Then calculating $g^{\frac{x^k}{\prod_{i=1}^{d}(x+m_i)}}$ for a single $k = 0, 1, 2, \ldots d$ takes $\Theta(d(T + T_p))$ time.*

Proof. We proceed as follows, for $0 \leq k < d$:

1. Calculate $m_1 = \xi^1$, $m_2 = \xi^2$, $\ldots$, $m_d = \xi^d$, and store these values. This calculation takes d multiplications.
2. Calculate $D_1 = \prod_{j=2}^{d}(m_j - m_1)$. This calculation takes $d - 1$ subtractions and $d - 2$ multiplications.
3. Calculate

$$D_2 = D_1 \cdot m_d \cdot \frac{1 - m_1}{\xi \cdot m_d - m_2}$$
$$D_3 = D_2 \cdot m_d \cdot \frac{1 - m_2}{\xi \cdot m_d - m_3}$$
$$\vdots$$
$$D_d = D_{d-1} \cdot m_d \cdot \frac{1 - m_{d-1}}{\xi \cdot m_d - m_d}$$

   Each row above takes 2 subtractions, 3 multiplications, and one division. There are $d - 1$ rows, so the total calculation takes $2(d - 1)$ subtractions, $3(d - 1)$ multiplications, and $d - 1$ divisions.
4. Define $e_i = \frac{(-m_i)^k}{D_i}$, for $i = 1$ to d. Calculate all of the e_i. Calculating a single e_i requires one exponentiation mod p and one division. Calculating all of the e_i's requires d exponentiations mod p and d divisions.
5. Define $f_i = \left(g^{1/(x+m_i)}\right)^{e_i}$, for $i = 1$ to d. Calculate all of the f_i. Calculating a single f_i requires one group exponentiation. Calculating all of the f_i requires d group exponentiations.
6. Calculate $g^{\frac{x^k}{\prod_{i=1}^{d}(x+m_i)}} = \prod_{i=1}^{d} f_i$. This calculation takes $d - 1$ group multiplications.

For $k = d$, we need to compute $g \cdot \prod_{i=1}^{d} f_i$ instead of $\prod_{i=1}^{d} f_i$ in the last step. The computation (and calculation of running time) is otherwise the same.

In the proof of Corollary 1, it is essential that the m_i's are chosen to equal ξ^i. Otherwise the relationship in Lemma 1 does not hold, and it is not obvious in this case how to calculate all of the D_i's using only $O(d)$ operations. A naive approach to calculating all the D_i's requires $O(d^2)$ multiplications and $O(d^2 T_p)$ total cost, which is how the $\Theta(dT + d^2 T_p)$ time in the statement of Proposition 1 was originally obtained in [16].

4.1 Security of the basic signature scheme

In this subsection, we show that weak existential forgery of the basic scheme under a chosen-message attack reduces to the q-Cheon-SDH problem. Like [16,

Theorem 4.3], Theorem 1 is a converse of [4, Lemma 9], but compared to [16, Theorem 4.3] and [4, Lemma 9], Theorem 1 features an improved reduction time of $\Theta(qT)$ to the q-Cheon-SDH problem, which is a harder problem, compared to $\Theta(q^2T)$ to the q-SDH problem for the prior results, using Corollary 1 to achieve this improvement.

Theorem 1. *If there is an algorithm that (t', ϵ')-breaks the q-Cheon-SDH problem, then we can (t, q_S, ϵ)-weakly break the basic Boneh-Boyen signature scheme provided that*

$$t \geq t' + \Theta(qT),\ q_S \geq q, \text{and } \epsilon \leq \frac{p-1-q}{p-1}\epsilon'$$

Proof. Let $\mathcal{A}$ be an algorithm that (t', ϵ')-breaks the q-Cheon-SDH problem. We will show that the adversary $\mathcal{B}$ can perform a weak existential forgery on the basic scheme of the signature under a chosen-message attack.

The adversary $\mathcal{B}$ begins by setting $m_i = \xi^i$ for $i = 1, \ldots, q_S$, where $q_S \geq q$, and $\xi \in \mathbb{Z}_p$ is chosen so that the m_i are distinct. Given a public key (g_1, g_2, g_2^x, ζ), along with corresponding valid signatures $(\sigma_1, \ldots, \sigma_{q_S}) = (g_1^{\frac{1}{x+m_1}}, \ldots, g_1^{\frac{1}{x+m_{q_S}}})$ for the m_i, the adversary $\mathcal{B}$ calculates $h_k \leftarrow g_1^{\frac{x^k}{\prod_{i=1}^{q}(x+m_i)}}$ for $k = 0, 1$, and q. Using Corollary 1, this calculation takes $O(qT)$ time. Then $\mathcal{B}$ chooses $\alpha \in \mathbb{Z}_p^*$ at random[4] and runs the algorithm $\mathcal{A}$ on inputs $(h_0^\alpha, h_1^\alpha, h_q^\alpha, g_2, g_2^x)$. Since this input is a valid q-Cheon-SDH instance, $\mathcal{A}$ outputs $(m_*, g_1^{\frac{\alpha}{(x+m_*)\prod_{i=1}^{q}(x+m_i)}})$ for some $m_* \in \mathbb{Z}_p$ with probability ϵ'. We then remove the α from the exponent in the second coordinate of the output by raising that second coordinate to the power of $\alpha^{-1} \bmod p$.

We note that the distribution of $(h_0^\alpha, h_1^\alpha, h_q^\alpha, g_2, g_2^x)$ above is identical to the distribution of uniformly random q-Cheon-SDH inputs, thanks to the inclusion of the random α. From the point of view of $\mathcal{A}$, the value of ξ does not influence the input distribution that it sees. Hence $\mathcal{A}$ has no better than random chance of choosing m_* to be one of $m_1, \ldots, m_q$. Thus, $m_* \neq m_i$ for all $i = 1, \ldots q$ with probability at least $\frac{p-1-q}{p-1}$. In the unlikely scenario that $m_* = m_i$ for some i, the algorithm $\mathcal{B}$ simply aborts. Otherwise, by Proposition 1,

$$\frac{1}{(x+m_*)\prod_{i=1}^{q}(x+m_i)} =$$
$$\frac{1}{(x+m_*)\prod_{i=1}^{q}(m_i-m_*)} + \sum_{j=1}^{q} \frac{1}{(x+m_j)\prod_{i\neq j}(m_i-m_j)}$$

Using this equation, $\mathcal{B}$ can calculate $\sigma_* = g_1^{\frac{1}{x+m_*}}$ as follows:

$$\sigma_* \leftarrow \left[g_1^{\frac{1}{(x+m_*)\prod_{i=1}^{q}(x+m_i)}} / \prod_{j=1}^{q} \sigma_j^{\prod_{i\neq j}(m_i-m_j)} \right]^{\prod_{i=1}^{q}(m_i-m_*)}$$

[4] In fact, simply using $\alpha = 1$ works well enough in practice. The only reason we use a random α is to randomize the input distribution, as explained in the next paragraph.

and then (m_*, σ_*) is an existential forgery for the basic scheme.

Note that calculating all of $\prod_{i \neq j}(m_i - m_j)$ takes time $O(qT_p)$, where T_p is the maximum time needed for one operation in $\mathbb{Z}_p$. Calculating all of $\sigma_j^{\prod_{i \neq j}(m_i - m_j)}$ takes time $O(qT)$. Calculating

$$\left[g_1^{\frac{1}{(x+m_*)\prod_{i=1}^{q}(x+m_i)}} / \prod_{j=1}^{q} \sigma_j^{\prod_{i \neq j}(m_i - m_j)} \right]$$

takes time $O(qT)$. Calculating, $\prod_{i=1}^{q}(m_i - m_*)$, takes time $O(qT_p)$. Given that $T_p \ll T$, and accounting for the t' time cost of $\mathcal{A}$, we find that $\mathcal{B}$ can perform a weak existential forgery in time t as long as $t \geq t' + \Theta(qT)$.

The proof above requires knowledge of g_1. This requirement is not a problem, since g_1 is published as part of the Boneh-Boyen public key. However, it is possible to construct a working variant of Boneh-Boyen in which g_1 is not published. In case g_1 is unknown, the theorem remains valid, provided that q is replaced by $q+1$ in the inequalities. In this case $\mathcal{B}$ uses $q+1$ signature queries and calculates $h'_k \leftarrow g_1^{\frac{x^k}{\prod_{i=1}^{q+1}(x+m_i)}}$ for $k = 0, 1$, and q in place of h_0, h_1, and h_q.

5 Cheon's Algorithm

Cheon [11,12] presents an algorithm which computes the secret exponent x from the input of an instance of the q-Cheon-SDH problem. Portions of this algorithm were also independently discovered by Brown and Gallant in the context of a different problem. In what follows, we refer to this algorithm as Cheon's algorithm. Specifically, Cheon proves the following:

Theorem 2. *Let $\mathbb{G}$ be a cyclic group of prime order p with g being a generator. Let T denote the maximum time needed for one exponentiation in $\mathbb{G}$.*

1. *Let d divide $p-1$. Given g, g^x, and g^{x^d}, the value of x can be recovered in time $O((\sqrt{p/d} + \sqrt{d})T)$.*
2. *Let d divide $p+1$. Given $g, g^x, g^{x^2}, \ldots, g^{x^{2d}}$, the value of x can be recovered in time $O((\sqrt{p/d} + d)T)$.*

Note that if $q \geq d$ in the first case or $q \geq 2d$ in the second case, then Cheon's algorithm can solve the q-SDH problem by revealing the secret exponent x. We show in this section that the algorithm can be applied to find the secret exponent in the basic scheme of the Boneh-Boyen signature over a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$.

Theorem 3. *Let T denote the maximum time needed to perform one group exponentiation in $\mathbb{G}_1$. Let $m_i = \xi^i$ for $i = 1, 2, \ldots, d+1$ or $i = 1, 2, \ldots, 2d+1$ respectively, where $\xi \in \mathbb{Z}_p$ and the m_i are distinct. Let d divide $p-1$. In the basic Boneh-Boyen scheme, if the adversary $\mathcal{A}$ queries for signatures of $m_i = \xi^i$ for $i = 1, 2, \ldots, d+1$, then the private exponent x can be computed in time $O((\sqrt{p/d} + d)T)$.*

Proof. Let d be a positive divisor of $p-1$. We construct an algorithm $\mathcal{A}$ which recovers the private exponent of the signature scheme under a chosen-message attack, using Cheon's algorithm. To start, $\mathcal{A}$ computes a primitive root ξ of $\mathbb{Z}_p$, which can be done by factoring $p-1$. Although factoring is usually treated as an expensive operation, in this case the cost of factoring is dominated by the cost of what follows. Given a public key (g_1, g_2, g_2^x, ζ), the algorithm $\mathcal{A}$ then queries for signatures for $m_1 = \xi, m_2 = \xi^2, ..., m_{d+1} = \xi^{d+1}$, upon which $\mathcal{A}$ obtains $d+1$ valid signatures $\sigma_1, \sigma_2, ..., \sigma_{d+1}$, where $\sigma_i = g_1^{\frac{1}{x+m_i}} = g_1^{\frac{1}{x+\xi^i}}$. Using Corollary 1, the algorithm $\mathcal{A}$ calculates

$$g_1^{\frac{1}{(x+m_1)\cdots(x+m_{d+1})}}, g_1^{\frac{x}{(x+m_1)\cdots(x+m_{d+1})}}, g_1^{\frac{x^d}{(x+m_1)\cdots(x+m_{d+1})}}$$

Then, it runs Cheon's algorithm in $\mathbb{G}_1$ with these inputs, and obtains x as output.

Since $g_1^{\frac{1}{(x+m_1)\cdots(x+m_{d+1})}}, g_1^{\frac{x}{(x+m_1)\cdots(x+m_{d+1})}}$ and $g_1^{\frac{x^d}{(x+m_1)\cdots(x+m_{d+1})}}$ together take time $\Theta(dT)$ to calculate, and Cheon's algorithm has a running time of $\Theta((\sqrt{p/d}+\sqrt{d})T)$, the overall running time is $\Theta((\sqrt{p/d}+d)T)$

6 Runtime Analysis

In this section, we will use Theorem 3 to compute the complexity, both experimentally and theoretically, of recovering the private key of a Boneh-Boyen signature for $d \mid p-1$. In this analysis, we only consider the case where g_1 is not included in the public key, which requires $d+1$ valid signatures. For convenience, we refer to the algorithm of Theorem 3 as the SDH algorithm.

6.1 Experimental Analysis

We implemented the SDH algorithm on 5 different Barreto-Naehrig curves with their corresponding p ranging from 30 to 50 bits. Barreto-Naehrig curves were chosen because they are suitable for pairing-based short signature schemes and easy to implement. We also implemented the Cheon's algorithm with the Pollard's kangaroo variant instead of the baby-step-giant-step variant or other variants such as Kozaki et al. [19], in order to save memory. All calculations were performed on an Intel(R) Xeon(R) Gold 6254 CPU at 3.10GHz.

The implementation of the SDH algorithm is straightforward. We wrote a program in SageMath and used functions in Sage's elliptic curve library to compute the triplet (g, g^x, g^{x^d}) using the method described in Corollary 1. The program then runs the Cheon's algorithm with the Pollard's kangaroo variant. In our implementation, the algorithm has 2 kangaroos, each having 9 possible steps in their random walk, with mean step size $\sqrt{p/d}$ in the first half of the algorithm. We also defined the distinguished points on the elliptic curve to constitute $\log(\frac{p}{d})/\sqrt{\frac{p}{d}}$ of all points, which is the optimal choice [13]. After two kangaroos collide, we can compute the value of k_0. In the second half of the algorithm, we can analogously compute the value of k_1 and the secret key $k = k_0 + k_1 \cdot \frac{p}{d}$.

For each curve, we performed at least 20 trials and the result taken is the average over all the trials. Although the optimal value of d is not always a divisor of $p-1$, we can always find nearby divisors and use them to estimate the hypothetical performance [14]. Figure 1 compares the measured performance of the SDH algorithm based on our chosen-message attack, with the generic Pollard's rho method for discrete log in which we simply input the public verification key into the generic Pollard's rho algorithm implementation built into Sage. Based on our results, the SDH algorithm outperforms Pollard's rho method when d is greater than 35. The absolute magnitude of the ratio between the two running times may not be as large as what is predicted by comparing $p^{1/2}$ and $p^{1/3}$, owing to the fact that our SDH kangaroo algorithm may not be as optimized as SAGE's built-in discrete log algorithm. What is most important, however, is that the limited data points available have slopes (on a log-log plot) consistent with the change in exponent from 1/2 to 1/3.

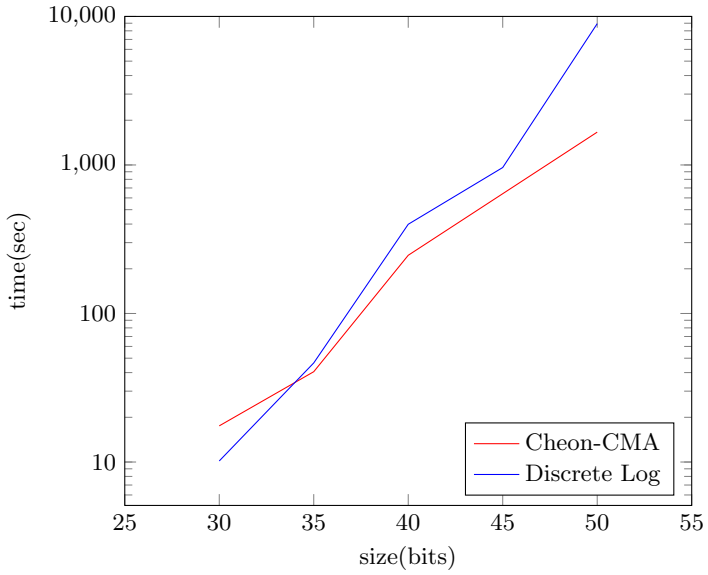

Fig. 1. Comparison of running times for our SDH-based chosen-message attack and generic Pollard's rho, on an Intel(R) Xeon(R) Gold 6254 CPU at 3.10GHz. Vertical axis represents the amount of wall-clock time required in order to recover the private signing key.

6.2 Theoretical Analysis

Now we calculate the theoretical cost of computing a Boneh-Boyen private key using the SDH algorithm. From Theorem 3, the SDH algorithm has a running

time of $O((\sqrt{p/d}+d)T)$, for the $p-1$ variant. This cost is minimized by taking $d = \Theta(p^{\frac{1}{3}})$, yielding a corresponding time complexity of $\Theta(p^{\frac{1}{3}} \cdot T) = \Theta(p^{\frac{1}{3}}(\log p) \cdot T_p)$ for the SDH algorithm. We remark that this time complexity matches the $\Omega(\sqrt[3]{p})$ generic lower bound proved in [4, Corollary 13]. Our result is the first to match this lower bound.

Our algorithm requires choosing a divisor d of $p-1$ which satisfies $d = \Theta(p^{\frac{1}{3}})$. In principle, such a divisor is not necessarily guaranteed to exist. However, in [16], the authors observe that a result of [14] implies that asymptotically all but a small proportion of random primes p admit divisors d for $p-1$ of a size suitable for our algorithm. Therefore, pairing-friendly curves are unlikely to resist our SDH algorithm unless specifically chosen with this property in mind.

7 Conclusion

In this paper, we showed that the existential forgery of a signature under a chosen-message attack for the basic Boneh-Boyen signature scheme can be reduced to the q-SDH problem via an algorithm with time complexity linear in q. This reduction represents an improvement from [16] and for the first time establishes an equivalence between the q-SDH assumption and the security of the basic Boneh-Boyen signature scheme which matches known generic lower bounds. Using Cheon's algorithm, the reduction allows us to recover the secret key used in Boneh-Boyen signatures in time $O(p^{\frac{1}{3}+\epsilon})$ for groups of order p whenever $p-1$ satisfies certain divisibility properties.

Jao and Yoshida in [16] also present a reduction for the case where we have a divisor d of $p+1$, using the second case of Theorem 2. Corollary 1 cannot be used in this case to obtain a faster reduction, since Corollary 1 doesn't asymptotically speed up the case where we need to compute an entire sequence of q-SDH inputs instead of just three q-Cheon-SDH inputs. We leave this case to future work.

Our results apply only to the basic Boneh-Boyen signature scheme. The authors in [16] also analyze the full Boneh-Boyen scheme, and show that a chosen-message attack on the full scheme can recover the private key in $O(p^{\frac{2}{5}+\epsilon})$ time. In this case there still remains a gap in complexity between the fastest known attack and the currently provable generic lower bound. It is not clear whether our approach can lead to improved attacks against the full scheme. The main obstacle is that in the full Boneh-Boyen scheme, the signer chooses a random integer r in order to randomize the signature result, and (assuming that the message m to be signed remains constant) this random integer r plays a role similar to the role played by m in the basic scheme. It is therefore easy to extend a known-message attack on the basic scheme to a chosen-message attack on the full scheme, as was done in [16]: simply "choose" to sign the same message repeatedly and then treat the resulting random r values as if they were known messages in the basic scheme. However, it is not easily possible to extend a chosen-message attack on the basic scheme, since one cannot control the random r values that the signer selects. For the time being, we leave this question to future work.

Acknowledgments. This work is supported by research funding from NSERC, CryptoWorks21, Public Works and Government Services Canada, Canada First Research Excellence Fund, and the Royal Bank of Canada.

References

1. Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The tower number field sieve. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology — ASIACRYPT 2015*, pages 31–55, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
2. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, pages 223–238, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
3. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, pages 56–73, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
4. Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, April 2008.
5. Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *J. Cryptology*, 24(4):659–693, October 2011.
6. Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 213–229, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
7. Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
8. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 514–532, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
9. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.
10. Daniel R. L. Brown and Robert P. Gallant. The static Diffie-Hellman problem. Cryptology ePrint Archive, Paper 2004/306.
11. Jung Hee Cheon. Security analysis of the Strong Diffie-Hellman problem. In Serge Vaudenay, editor, *Advances in Cryptology — EUROCRYPT 2006*, pages 1–11, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
12. Jung Hee Cheon, Taechan Kim, and Yongsoo Song. *The discrete logarithm problem with auxiliary inputs*, pages 71–92. De Gruyter, 2014.
13. Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2005.

14. Kevin Ford. The distribution of integers with a divisor in a given interval. *Annals of Mathematics*, 168(2):367–433, 2008.
15. Aurore Guillevic. A short-list of pairing-friendly curves resistant to special tnfs at the 128-bit security level. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography — PKC 2020*, pages 535–564, Cham, 2020. Springer International Publishing.
16. David Jao and Kayo Yoshida. Boneh-Boyen signatures and the Strong Diffie-Hellman problem. In Hovav Shacham and Brent Waters, editors, *Pairing-Based Cryptography — Pairing 2009*, pages 1–16, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
17. Neal Koblitz and Alfred Menezes. Another look at generic groups. *Advances in Mathematics of Communications*, 1(1):13–28, 2007.
18. Neal Koblitz and Alfred Menezes. The brave new world of bodacious assumptions in cryptography. *Notices of the American Mathematical Society*, 57(3):357–365, 2010.
19. Shunji Kozaki, Taketeru Kutsuma, and Kazuto Matsuo. Remarks on Cheon's algorithms for pairing-related problems. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Pairing-Based Cryptography — Pairing 2007*, pages 302–316, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
20. Claus P. Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 239–252, New York, NY, 1990. Springer New York.
21. Claus P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, January 1991.