

The Exact Security of Pairing Based Encryption and Signature Schemes

David Galindo

Departament Matemàtica Aplicada IV
Universitat Politècnica de Catalunya
Campus Nord, c/Jordi Girona, 1-3, 08034 Barcelona
dgalindo@mat.upc.es

<p>Comment: These notes are intended to be a complement to the talk of the same title in the Workshop on Provable Security to be held at INRIA, 3-5 November 2004</p>
--

Working Draft
November 1, 2004

Abstract. Bilinear pairings have been intensively used in the design of cryptographic protocols during the last few years. For instance, short signatures and non-interactive key exchange protocols have been designed with them, and they appear as a key component for identity-based cryptography.

Focusing on encryption and signature schemes built on bilinear pairings, we look at the security reductions of some known constructions. For any pair “scheme/security reduction”, we deduce key sizes to securely implement the schemes. It turns out that some important protocols in the literature appear to be not as efficient as one would wish, due to the lack of tightness of their security reductions. Finally, we summarize current trends to obtain tight security reductions and suggest some open problems.

Keywords: encryption, signatures, provable security, exact security, bilinear maps.

1 Introduction

Practice oriented provable security [Bel98] was introduced in a set of papers authored by Bellare and Rogaway. The idea was to explicitly capture the quantitative aspects of security, by means of an exact treatment of the security reductions. This would allow to obtain practical measurements such as the number of cycles of adversary computation the scheme can withstand or how long the security parameter is. Another of the goals pursued is to preserve as much as possible the strength of the underlying hard problem which is used in the protocol’s design. Thus, one looks for tight reductions, since this directly affects

the efficiency of the protocol. A non-tight reduction means that to get the same security level we must use larger keys for the primitive we are using, and this means slower protocols. We will indistinctly refer to these ideas as the practice oriented security or exact security approaches.

Since then, finding tight reductions under trusted assumptions for cryptographic protocols has been a very active and fruitful line of research. It turns out that most of the efforts for achieving tight or improved reductions are for constructions that use the Random Oracle heuristic (ROM) [BR93] in their security proofs.

On the other hand, in the last few years bilinear maps and bilinear group pairs [BLS01] have been found a lot of appealing applications [Jou00,Bar]. Most of these new protocols use the ROM in their security study and present a non-tight security reduction. Although the search for improved security reductions for these protocols has been directly addressed in the literature (for instance in [KW03,LQ04]), works taking into account practical implications such as the key size or the efficiency of the schemes are rarely found. Even more, in some key works the security parameters are proposed as if the security reductions were tight, when they are not [BF01,BLS01].

Maybe one of the reasons behind this can be found in the following quotation from two of the currently leading researchers in these matters:

“We note, however, that these proofs are set in the random oracle model and therefore it is not clear whether the efficiency of the security reduction is relevant to actual security in the real world.”¹

Our contributions. In this work we address the issue of the non-tight security reductions for some encryption and signature schemes using bilinear maps. In the first place, we study some implications of this lack of tightness inside the practice oriented security approach. We do so by looking at two of the pioneering works in this field [BF01,BLS01], which deal with the issues of identity-based encryption and short signatures respectively.

Our first goal is to obtain the security parameters related to the security reductions shown by the authors. We do it in a rigorous fashion, following the relevant paper [LV01] to set up the computational setting. As a consequence, the efficiency of the encryption scheme [BF01] is seriously affected; also the length for the signatures in [BLS01] is increased in such a way that they do not meet the short signatures goal pursued by the authors. We stress that these negative results are obtained if we rigorously use the exact security approach.

Secondly, we investigate how to solve these undesired results when taking into account security reductions in the ROM. The idea is to try to balance between practice-oriented security and efficient schemes, in such a way that we can accept that a protocol is implemented with shorter keys than it should be, but we would like to be shown an argument supporting this decision. In this way, we show improved security reductions with respect to the assumptions used

¹ This comment is found in [BB04], page 69 of the proceedings version and in page 14 of the electronic version from the authors web page.

in the original works, and suggest also new (possibly stronger but plausible) assumptions.

2 Exact security preliminaries

Criterion 1 (Concrete security). The efficiency of a security reduction for an encryption/signature scheme is the relationship between:

- an *attacker* who breaks an encryption/signature scheme with advantage at least ε in time t , doing less than q_D calls to a decryption oracle or q_S calls to a signing oracle, and less than $q_{\mathcal{O}_i}$ calls to some random oracles \mathcal{O}_i ; and
- the implied (t', ε') *solver* against the corresponding trusted cryptographic assumption, running in time at most t' and with advantage at least ε' .

Such an attacker is referred as a $(t, q_D, q_{\mathcal{O}_i}, \varepsilon)$ or $(t, q_S, q_{\mathcal{O}_i}, \varepsilon)$ attacker for short. Following the usual terminology, the security reduction is *tight* if $\frac{t'}{\varepsilon'} \approx \frac{t}{\varepsilon}$, and *not tight* if $\frac{t'}{\varepsilon'} > q_{\mathcal{O}} \frac{t}{\varepsilon}$, where $q_{\mathcal{O}} \in \{q_D, q_S, q_{\mathcal{O}_i}\}$. It is also said that a scheme is *very tight* if $\varepsilon \approx \varepsilon'$ and t' is equal to t plus a linear quantity in the number of oracle calls. The tighter is the reduction, the smaller is the gap between the computational efforts needed to break the scheme and to solve the underlying problem. The optimal tightness is achieved with *very tight* reductions.

Criterion 2 (Security level). We use most of the considerations in the work [LV01] by Lenstra and Verheul, which is an important reference for the determination of secure key sizes for the most usual cryptographic primitives. A standard approach for defining the security level computational or difficulty of a problem is the following: a problem \mathcal{P} is said to have security level 2^t when the minimal computational effort to solve this problem should be of the order of 2^t 3-DES encryptions [X9.01]. Following this terminology, the current demanded security level is 2^{80} . This was for instance the requirement for the asymmetric primitives submitted to the NESSIE project [NES03]. In [LV01] is argued that this computational effort can be assumed infeasible at least until the year 2013 (and probably some years onwards, since this is a lower bound estimation).

Criterion 3 (Time units). We need to fix the time units we will use in order to relate key sizes and security reductions. In our setting, the speeds of a hash function application and a DES as well as 3-DES encryption are considered comparable. They represent our time unit to match the definition of security level. In order to establish equivalences with single operations on a PC, we need to figure out how many single PC operations can be performed in a time unit. For instance, 360 [Bih97] and 500 Pentium clock cycles [Sch] have been estimated for software implementations of DES. We use the latter value with the aim to balance the fact that we have considered 3-DES and DES to have comparable speed. We stress that these choices are consistent with [LV01] and similar works. Therefore, throughout this work

1 time unit = 500 Pentium clock cycles.

Criterion 4 (Modular multiplication cost). In [LV01], the authors report that their own experiments show that a multiplication in a field of size k takes about $k^2/24$ clock cycles, that is, $k^2/(24 \cdot 500)$ time units.

Criterion 5 (Practical computational complexity). Let $\mathcal{P}_1, \mathcal{P}_2$ two problems such that there exists a reduction from \mathcal{P}_2 to \mathcal{P}_1 but the existence of the opposite reduction is unknown. If in addition the only known way to solve \mathcal{P}_2 is to solve \mathcal{P}_1 , it is generally *assumed* that both problems have similar complexity from a practical point of view. This criterion is used for the computation of secure key sizes for cryptographic schemes.

3 Bilinear maps and bilinear group pairs

Most of the material in this section resembles [BLS04b,PSV04]. Let $\mathbb{G}_1, \mathbb{G}_2$ and \mathbb{G}_T be finite abelian groups in which the discrete logarithm is believed to be hard. We use additive notation for $\mathbb{G}_1, \mathbb{G}_2$ whereas multiplicative notation is used for \mathbb{G}_T . By a *pairing* or *bilinear map* we will refer to a non-degenerate bilinear function $\hat{t} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. In some protocols the existence of a computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is assumed. Bilinear maps are usually implemented using the Weil or modified Tate pairings on an elliptic curve. The Weil pairing was introduced in cryptography in [MOV93] to attack the elliptic curve logarithm problem over certain curves, while the Tate pairing was introduced in [FR94], in an extension of the former work. Let $\mathbb{G}_1 = E(\mathbb{F}_q)$ denote an elliptic curve over the finite field \mathbb{F}_q with order divisible by p such that p also divides $q^\alpha - 1$, where α is the order of q in \mathbb{Z}_p^* and is called the MOV embedding degree. The modified Tate pairing $\hat{t}(\cdot, \cdot)$, which is the bilinear map usually recommended, takes values in the subgroup \mathbb{G}_T of $\mathbb{F}_{q^\alpha}^*$ of order p and is defined in two ways, depending on whether E is a supersingular or ordinary curve.

In the supersingular case $\mathbb{G}_1 = \mathbb{G}_2 = E(\mathbb{F}_q)$, and there exists a so-called *distorsion map* $\phi : E(\mathbb{F}_q) \rightarrow E(\mathbb{F}_{q^\alpha})$. In this case, the modified Tate pairing is proven non-degenerate, and in addition for all $P, Q \in E(\mathbb{F}_q)$ we have

$$\hat{t}(P, Q) = \hat{t}(Q, P).$$

This implies that the groups \mathbb{G}_1 and \mathbb{G}_2 can be treated equally when designing cryptographic protocols. For supersingular curves the best parameter choices are in characteristic 3, and in this case $\alpha = 6$ can be selected (which is the maximum possible value for α in these curves).

Ordinary curves can also be used to implement pairings, but in this case \mathbb{G}_2 is set to be a subgroup of $E(\mathbb{F}_{q^\alpha})$. Here distorsion maps are not available, and it is no longer possible to exchange the roles of the groups \mathbb{G}_1 and \mathbb{G}_2 when defining a protocol. Several techniques have been presented [BKLS02,BLS04a] in order to improve efficiency and bandwidth. For instance, the twist of the curve $E(\mathbb{F}_{q^{\alpha/2}})$ can be used to generate the group \mathbb{G}_2 , and therefore there is a 50%

save in bandwidth.

Regarding the existence of the isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$, in [BLS01] is shown that ψ naturally exists in all the group pairs we considered above. Usually, \mathbb{G}_1 and \mathbb{G}_2 are set to be cyclic groups of prime order p , and $\psi(P_2) = P_1$, where P_1 and P_2 generate $\mathbb{G}_1, \mathbb{G}_2$ respectively. When implemented with the Weil or modified Tate pairings, \mathbb{G}_T is a p order subgroup of \mathbb{F}_q^* . With this setup we obtain natural generalizations of the CDH and DDH problems. In the following $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$, where p is prime; $\mathbb{G}_1, \mathbb{G}_2$ are cyclic groups generated by P_1, P_2 respectively and $\psi(P_2) = P_1$.

Computational Diffie-Hellman problem on $(\mathbb{G}_1, \mathbb{G}_2)$. Given $P_2, aP_2 \in \mathbb{G}_2$ and $Q \leftarrow \mathbb{G}_1$ as input, compute $aQ \in \mathbb{G}_1$, where $a \leftarrow \mathbb{Z}_p^*$.

We say an algorithm \mathcal{B} is (t, ε) breaking CDH on $(\mathbb{G}_1, \mathbb{G}_2)$ if it runs in time at most t and has advantage at least ε , where the probability is over the random choices of the parameters, and the random bits of \mathcal{B} .

Decisional Diffie-Hellman problem on $(\mathbb{G}_1, \mathbb{G}_2)$. Given $P_2, aP_2 \in \mathbb{G}_2$ and $Q, bQ \leftarrow \mathbb{G}_1$ as input, output **yes** if $a = b$ and **no** otherwise, where $a \leftarrow \mathbb{Z}_p^*$.

Bilinear Diffie-Hellman (BDH) Problem on $(\mathbb{G}_1, \mathbb{G}_2)$. Given $P_2, cP_2 \in \mathbb{G}_2^*$ and $aP_1, bP_1 \in \mathbb{G}_1^*$, where $P_2 \leftarrow \mathbb{G}_2^*$, $P_1 = \psi(P_2)$, $a, b, c \leftarrow \mathbb{Z}_p^*$; compute $W = \hat{t}(P_1, P_2)^{abc} \in \mathbb{G}_T$.

We say an algorithm \mathcal{B} is (t, ε) breaking BDH on $(\mathbb{G}_1, \mathbb{G}_2)$ if it runs in time at most t and has advantage at least ε , where the probability is over the random choices of the parameters, and the random bits of \mathcal{B} .

Remark 1. Due to the computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$, it is indifferent if we are given the value xP_1 or xP_2 in the BDH input, since $\psi(xP_2) = xP_1$. We also note that the usual definition of the BDH problem in the literature uses $\mathbb{G}_1 = \mathbb{G}_2$ and $\langle P_1, aP_1, bP_1, cP_1 \rangle$ as the input.

Decision Bilinear Diffie-Hellman (DBDH) Problem on $(\mathbb{G}_1, \mathbb{G}_2)$. Given $P_2, cP_2 \in \mathbb{G}_2^*$, $aP_1, bP_1 \in \mathbb{G}_1^*$, and $T \in \mathbb{G}_T$, where $P_2 \leftarrow \mathbb{G}_2^*$, $P_1 = \psi(P_2)$, $a, b, c \leftarrow \mathbb{Z}_p^*$ and $T \leftarrow \mathbb{G}_T$; output **yes** if $T = \hat{t}(P_1, P_2)^{abc}$ and **no** otherwise.

We say an algorithm \mathcal{B} is (t, ε) breaking DBDH on $(\mathbb{G}_1, \mathbb{G}_2)$ if it runs in time at most t and has advantage at least ε , where the probability is over the random choices of the parameters, and the random bits of \mathcal{B} .

Definition 1 (Bilinear group) *Two prime p order groups $(\mathbb{G}_1, \mathbb{G}_2)$ are a (t, ε) -bilinear group pair if they satisfy the following properties:*

- *The group operation on both \mathbb{G}_1 and \mathbb{G}_2 and the map $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ can be computed in polynomial time.*
- *A group \mathbb{G}_T of order p and a non-degenerate bilinear map $\hat{t} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ exist, and \hat{t} is computable in polynomial time.*
- *No algorithm (t, ε) breaks CDH on $(\mathbb{G}_1, \mathbb{G}_2)$.*

Informally, we are interested in bilinear group pairs where p is sufficiently small so that the map \hat{t} is efficiently computable, but t/ε is sufficiently large so that CDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is intractable.

The best known algorithm for solving CDH on $(\mathbb{G}_1, \mathbb{G}_2)$ is to compute discrete log on \mathbb{G}_1 . As a consequence of the map \hat{t} , computing discrete logarithms in \mathbb{G}_1 can be transformed into computing the discrete logarithm in the subgroup \mathbb{G}_T . The fastest method for solving DL on a random elliptic curve is the Pollard's parallelizable ρ method [Pol78], which runs in exponential time equivalent to $0.88\sqrt{p}$ group operations in \mathbb{G}_1 . Assuming that $p \approx q$, an addition in $E(\mathbb{F}_q)$ lasts for $12 \cdot \log^2 p / (24 \cdot 500)$ time units. This is obtained assuming that an addition in $E(\mathbb{F}_q)$ takes 12 multiplications in the underlying field. Then, assuming that DL in \mathbb{G}_1 has the same complexity as DL in a random curve,

$$\left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_1} \approx \frac{0.88 \cdot 12 \cdot \sqrt{p} \log^2 p}{24 \cdot 500} = 0.88 \frac{\sqrt{p} \log^2 p}{10^3} \text{ time units.}$$

In the case of \mathbb{G}_T , there exists an index calculus algorithm for discrete logarithm running in subexponential time. The expected number of single operations is proportional to $e^{c^{1/3} \ln^{1/3} q^\alpha \ln^{2/3}(\ln q^\alpha)}$, where $c = 64/9$ in large characteristic fields and $c = 32/9$ in low characteristic fields. Dividing the resulting quantity by 500 we obtain the number of time units required to solve DL in \mathbb{G}_T . Therefore,

$$\left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_T} \approx \frac{e^{c^{1/3} \ln^{1/3} q^\alpha \ln^{2/3}(\ln q^\alpha)}}{500} \text{ time units.}$$

Lenstra and Verheul suggest the above estimation must be adjusted because it is reasonable to expect that cryptanalytic advances will become twice as effective computing DL in $\mathbb{F}_{q^\alpha}^*$ each 18 months. In the case of the security level 2^{80} this means to divide by 645 (cf. [LV01]), that is

$$\left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_T}^{\text{LV}} \approx \frac{e^{c^{1/3} \ln^{1/3} q^\alpha \ln^{2/3}(\ln q^\alpha)}}{500 \cdot 645} \text{ time units.}$$

Joux and Nguyen [JN03] showed that a bilinear map \hat{t} provides an algorithm for solving the DDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$. If $(\mathbb{G}_1, \mathbb{G}_2)$ is a bilinear group pair, then (P_2, V, Q, σ) is a DH tuple if and only if $\hat{t}(Q, V) = \hat{t}(\sigma, P_2)$.

4 Boneh-Lynn-Shacham short signature scheme

In this section we study some security reductions related to the BLS signature scheme [BLS01]. We begin by describing the protocol. Let $(\mathbb{G}_1, \mathbb{G}_2)$ a (t, ε) -bilinear group pair as in definition 1. Let $H : \{0, 1\}^* \rightarrow \mathbb{G}_1$ be a full-domain hash function. A signature σ is an element in \mathbb{G}_1 . The BLS signature scheme

comprises the following protocols:

KeyGeneration. Pick a random $x \leftarrow \mathbb{Z}_p^*$ and compute $V = xP_2$. The public key is $V \in \mathbb{G}_2$. The private key is x .

Signing. Given a private key $x \in \mathbb{Z}_p^*$, and a message $M \in \{0, 1\}^*$, compute $Q = H(M) \in \mathbb{G}_1$ and $\sigma = xQ$. The signature is $\sigma \in \mathbb{G}_1$.

Verification. Given a public key $V \in \mathbb{G}_2$, a message $M \in \{0, 1\}^*$ and a signature $\sigma \in \mathbb{G}_1$, compute $Q = H(M) \in \mathbb{G}_1$ and verify that (P_2, V, Q, σ) is a valid DH tuple. If so, output **valid**; otherwise output **invalid**.

For the definition of existential unforgeability under chosen message attacks see [BLS01]. The authors prove the following result in the Random Oracle Model [BR93].

Theorem 2 *Let $(\mathbb{G}_1, \mathbb{G}_2)$ a (t', ε') -bilinear group pair of order p . Then the signature scheme on $(\mathbb{G}_1, \mathbb{G}_2)$ is $(t, \varepsilon, q_S, q_H, \varepsilon)$ -secure against existential forgery under and adaptive chosen message attack (in the random oracle model), for all t and ε satisfying*

$$t \leq t' - c_{\mathbb{G}_1}(q_H + 2q_S) \quad \text{and} \quad \varepsilon \geq e(q_S + 1) \cdot \varepsilon'.$$

Here $c_{\mathbb{G}_1}$ is the time for computing an exponentiation in \mathbb{G}_1 and e is the base of the natural logarithm.

4.1 Computing the security parameter using the original reduction

Let us assume that the existential unforgeability under chosen message attacks of BLS scheme is $(t, q_S, q_H, \varepsilon)$ -broken by some adversary \mathcal{A} . Since this adversary can be run repeatedly (with the same input and independent internal coin tosses), the expected time to produce a forgery is t/ε . Thus, the security parameter of the scheme is $n_{\text{BLS}} = \log(t/\varepsilon) = n + m$, where $n = \log t$ and $m = \log(1/\varepsilon)$. In Theorem 2 is shown that such an adversary can be used to break CDH in $(\mathbb{G}_1, \mathbb{G}_2)$ within time at most t' and probability at least ε' in the ROM, where

$$t' \approx t + c_{\mathbb{G}_1}(q_H + 2q_S) \quad \text{and} \quad \varepsilon' \leq \frac{\varepsilon}{e(q_S + 1)}.$$

Usually, $q_S \leq 2^{30}$ (that is, up to one billion signature queries are allowed), and $q_H \leq t = 2^{60}$. We assume that computing a multiple in \mathbb{G}_1 requires in the worst case $\mathcal{O}(\log q)$ multiplications in \mathbb{F}_q . Since a multiplication in a field of size $\log q$ takes about $\log^2 q / (24 \cdot 500)$ time units (see criterion 4), we obtain that computing an exponentiation in \mathbb{F}_q requires $\log^3 q / (24 \cdot 500)$ time units. Therefore,

$$t' \approx t + 2^{60} \cdot \frac{\log^3 q}{24 \cdot 500} + 2^{31} \cdot \frac{\log^3 q}{24 \cdot 500}$$

Setting $m = 0$ and $n = 80$, we obtain $n_{\text{BLS}} = 80$, that is, a 2^{80} security level for BLS scheme. That is,

$$t' \approx 2^{80} + 2^{60} \cdot \frac{\log^3 q}{24 \cdot 500} \quad \text{and} \quad \varepsilon' \leq \frac{1}{e \cdot 2^{30}},$$

neglecting small terms. The latter reduction is meaningful, for the values of $\log q$ such that is more efficient to attack the scheme using the adversary \mathcal{A} than directly computing discrete logarithms in \mathbb{G}_1 . Assuming that $p \approx q$, the minimal value for $\log q$ is the smallest value k such that

$$\frac{t'}{\varepsilon'} \leq \left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_1}, \quad \text{that is,} \quad 2^{110} + 2^{90} \cdot \frac{\log^3 q}{24 \cdot 500} \leq \frac{0.88}{e} \sqrt{q} \frac{\log^2 q}{10^3}.$$

Numerically solving this inequality we obtain that $k = 213$. If we make the *assumption* that the CDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ and the DL problem on \mathbb{G}_1 have comparable computational difficulty, then a group \mathbb{G}_1 with $213 \leq |p|$ is needed to get a 2^{80} security level in BLS scheme. This is not enough, however. The reason is that we must also check that is better to attack the scheme using the adversary \mathcal{A} than directly computing discrete logarithms in \mathbb{G}_T . For supersingular curves we study the case $\alpha = 6$, while for ordinary curves we study $\alpha = 6, 10$. In general, the inequality involved is

$$\frac{t'}{\varepsilon'} \approx e \cdot 2^{110} + e \cdot 2^{90} \cdot \frac{\log^3 q}{24 \cdot 500} \leq \left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_T} \quad \text{or} \quad \left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_T}^{\text{LV}}$$

where $c = 64/9$ in large characteristic fields and $c = 32/9$ in low characteristic fields. After doing the corresponding calculations, one obtains that in the supersingular case $|q| \geq 636$ (761); in the ordinary case $|q| \geq 368, 221$ (438, 263) for $\alpha = 6, 10$ respectively, where in brackets appears the values following [LV01].

In [BLS04b] the exact security analysis seems to be not taken into account, and they propose to use ordinary curves with $\alpha = 6$, $p \approx q$ and $|q| \geq 171$ to implement their signature scheme. However, following the exact security approach as exemplified in [BR96] there is no guarantee that for this security parameter the scheme is secure (however this *does not mean* there is an attack against the scheme). Our estimation leads to $|p| \geq 214$ and $|q| \geq 368$ or 438, which in the case of q is roughly 1.5 or 2 times the value previously proposed. As a consequence, the signatures obtained are not shorter than DSA signatures, which was the goal pursued by the authors.

4.2 Katz&Wang modification allowing a tight security reduction

The reason for the factor q_S in the probability of forgery in Theorem 2 is the same that in the case of the full-domain-hash (FDH) signature scheme [BR93] when implemented with RSA. In fact, the reduction of Theorem 2 resembles Coron's analysis in [Cor00]. What happens in both cases is the following: since each message has a unique signature, for each M the simulator has two choices,

- it is able to produce a signature of M , in which case a forged signature does not provide any useful information;

- it is not able to produce a signature of M . In this case, a forged signature of m allows to break the underlying assumption, but a signing query for this message causes the simulator to abort.

This motivated the design of probabilistic signature schemes such as PSS [BR96], in which each message has multiple signatures, and this time a tight security reduction is obtained. Recently, Katz and Wang [KW03] figured out that is possible to get a tight security reduction without probabilistic signing. Their scheme has the property that each message has two valid signatures, but only one valid signature is provided by the signer. This property allows the simulator to correctly answer *all* signing queries. This idea can be applied to BLS scheme, providing a deterministic signature scheme with a tight reduction. The new scheme can be described as follows:

ModifiedBLS signature scheme

The global parameters are the same as in BLS scheme, except for a new hash function $G : \{0, 1\}^* \rightarrow \{0, 1\}$.

NewKeyGeneration. The same as **KeyGeneration** in BLS scheme. The public key is $V \in \mathbb{G}_2$. The private key is $x \in \mathbb{Z}_p^*$.

NewSigning. Given a private key $x \in \mathbb{Z}_p^*$, and a message $M \in \{0, 1\}^*$, compute $b = G(x, M)$, $Q = H(b, M) \in \mathbb{G}_1$ and $\sigma = xQ$. The signature is $(b, \sigma) \in \{0, 1\} \times \mathbb{G}_1$.

NewVerification. Given a public key $V \in \mathbb{G}_2$, a message $M \in \{0, 1\}^*$ and a signature $(b, \sigma) \in \{0, 1\} \times \mathbb{G}_1$, compute $Q = H(b, M) \in \mathbb{G}_1$ and verify that (P_2, V, Q, σ) is a valid DH tuple. If so, output **valid**; otherwise output **invalid**.

Following the proof of [Theorem 2, KW03], the following holds

Result 3 *Let $(\mathbb{G}_1, \mathbb{G}_2)$ a (t', ε') -bilinear group pair of order p . Then the signature scheme on $(\mathbb{G}_1, \mathbb{G}_2)$ is $(t, q_S, q_H, \varepsilon)$ -secure against existential forgery under and adaptive chosen message attack (in the random oracle model), for all t and ε satisfying*

$$t \leq t' - c_{\mathbb{G}_1}(q_H + 2q_S) \quad \text{and} \quad \varepsilon \geq 2 \cdot \varepsilon'.$$

Now, adapting our discussion in Section 4.1 to this particular case, we have

$$\frac{t'}{\varepsilon'} \approx 2^{81} + 2^{61} \cdot \frac{\log^3 q}{24 \cdot 500}.$$

Again, this value is meaningful from a security point of view when

$$\frac{t'}{\varepsilon'} \leq \left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_1} \quad \text{and} \quad \frac{t'}{\varepsilon'} \leq \left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_T} \quad \text{or} \quad \left(\frac{t}{\varepsilon}\right)_{\mathbb{G}_T}^{\text{LV}}.$$

When computing the key size for a 2^{80} security level, the corresponding calculations leads to $|p| \geq 154$ and $|q| \geq 194$ ($|q| \geq 242$ with the LV criterion) for ordinary curves with $\alpha = 6$. Since the ModifiedBLS scheme has a tight security reduction, shorter signatures than DSA signatures are obtained and with security guarantees (in the sense of the exact security approach [BR96]).

4.3 A tight reduction for BLS under a new (reasonable) assumption

Let us summarize what we have seen with respect to BLS signature up to this moment. On the one hand, the exact security approach leads to key sizes not as short as proposed in [BLS01]. On the other hand, the slight BLS modification in [KW03] presents a tight security reduction. Therefore, this new scheme provides short signatures within the practice-oriented provable security approach. If we want short signatures, should we use ModifiedBLS instead of BLS scheme? The answer is yes if we rigorously follow the exact security framework. But we stress that there is *no attack* known against BLS scheme when implemented with the parameters proposed by their authors. Then, can we use BLS scheme as if it had a tight security reduction? Can we identify what this mean in cryptographic terms?

To deal with these questions, we use some ideas from Koblitz and Menezes in [KM04], where they give an informal analysis and critique of several important aspects of the provable security methodology. In [KM04,Section 3] they discuss the security of FDH [BR93] and PSS [BR96] signature schemes for the RSA case. As we said before, BLS scheme can be seen as a FDH-like signature scheme. In [KM04] an RSA-related problem is defined (which is called RSA1 problem), arguing (using our terminology) that then FDH scheme has a tight reduction under this new problem. It turns out that this idea can be extended to the BLS scheme. Let us define the new CDH-related problem on $(\mathbb{G}_1, \mathbb{G}_2)$.

Definition 4 (CDH1(q_S, q_H) problem on $(\mathbb{G}_1, \mathbb{G}_2)$) *Given a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$, with $\mathbb{G}_2 = \langle P_2 \rangle$; aP_2 , where $a \leftarrow \mathbb{Z}_p^*$ is unknown; and a set of $q_S + q_H$ values Q_i chosen at random in \mathbb{G}_1 , you are allowed (at whatever stages of your work that you choose) to select up to q_S of those Q_i for which you'll be given the solutions aQ_i . You must produce a solution $aQ_j \in \mathbb{G}_1$ for one of the remaining Q_j .*

Result 5 *Let $(\mathbb{G}_1, \mathbb{G}_2)$ a bilinear group pair in which there is no algorithm (t', ε') breaking the CDH1(q_S, q_H) problem. Then the BLS signature scheme on $(\mathbb{G}_1, \mathbb{G}_2)$ is $(t, q_S, q_H, \varepsilon)$ -secure against existential forgery under an adaptive chosen message attack (in the random oracle model), where $t \approx t'$ and $\varepsilon \approx \varepsilon'$.*

Proof. Let \mathcal{A} a $(t, q_S, q_H, \varepsilon)$ forger of the signature scheme. We construct an algorithm \mathcal{B} breaking CDH1(q_S, q_H) problem within essentially the same time t and the same advantage ε . \mathcal{B} receives from its challenger $P_2 \leftarrow \mathbb{G}_2$ and $aP_2 \in \mathbb{G}_2$, where $a \leftarrow \mathbb{Z}_p^*$, and an ordered list $Q_{\text{list}} = \{Q_1, \dots, Q_{q_S+q_H}\}$ of uniformly independent random elements from \mathbb{G}_1 . Algorithm \mathcal{B} simulates the challenger to \mathcal{A} .

Setup. \mathcal{A} receives the generator P_2 and the public key aP_2 .

H-queries. At any time algorithm \mathcal{A} can query the random oracle H . To answer these queries \mathcal{B} maintains a list of entries $(M_i, Q_i) \in \{0, 1\}^* \times \mathbb{G}_1$ as explained next. When \mathcal{A} queries the oracle with a message M_i , algorithm \mathcal{B} responds as follows:

1. If the query M_i appears on the H -list in an entry (M_i, Q_i) the algorithm \mathcal{B} returns $H(M_i) = Q_i$.
2. Otherwise, \mathcal{B} takes the first element in the list Q_{list} , which is referred to as Q_i and adds (M_i, Q_i) to the H -list. Finally, \mathcal{B} responds to \mathcal{A} with $H(M_i) = Q_i$ and removes the first element from Q_{list} .

Signature queries. When \mathcal{A} asks for a signature of a message M_i , algorithm \mathcal{B} responds as follows: Algorithm \mathcal{B} runs the above algorithm for answering H -queries to obtain $Q_i \in \mathbb{G}_1$. Let (M_i, Q_i) the corresponding entry in H -list. Then \mathcal{B} queries its challenger with Q_i and gets back aQ_i . Finally, answers \mathcal{A} with $\sigma_i := aQ_i$.

Output. Eventually algorithm \mathcal{A} produces a signature-message pair (M_j, σ_j) such that M_j was not previously signed by \mathcal{B} . Then \mathcal{B} gives (Q_j, σ_j) to its challenger as a solution to the $\text{CDH1}(q_S, q_H)$ problem. Since the simulation is perfect, t' is essentially equal to t and $\varepsilon' = \varepsilon$. \square

Let us discuss the hardness of the CDH1 problem. Obviously, if you can solve CDH on $(\mathbb{G}_1, \mathbb{G}_2)$, you can also solve CDH1 with the same advantage and roughly within the same time. The converse also holds when $q_S = 0$. The best reduction known from CDH to CDH1 in the general case, which is essentially obtained translating the original BLS security proof to this terminology, states that CDH can be solved in time of order $\mathcal{O}(q_S)$ times the amount of time needed to solve CDH1. Therefore, CDH and CDH1 problem are equivalent using a reduction argument, but *not* tightly equivalent. So the problem of using BLS scheme with short keys have not been solved yet. To this aim, we have to look at CDH1 problem from a practical point of view. Let us state the following

Conjecture 6 *The problems CDH and $\text{CDH1}(q_S, q_H)$ on $(\mathbb{G}_1, \mathbb{G}_2)$ are indistinguishable in practice, that is, they have similar computational complexity.*

In other words, to claim that BLS scheme is secure with small key sizes is *equivalent* from an exact security point of view to claim that conjecture 6 holds. But, is it a reasonable conjecture? As maintained by Kobitz and Menezes for a similar problem with respect to RSA, we think that in this context is a reasonable assumption. In fact, when computing key sizes for pairing based schemes, we are already accepting at least two computational equivalences:

Assumption I. CDH on a random curve and CDH on \mathbb{G}_1 , where $(\mathbb{G}_1, \mathbb{G}_2)$ is a bilinear group pair, have similar complexity (whenever DL is secure in \mathbb{G}_T).

Assumption II. DL and CDH on a random curve have similar computational complexity.

Assumption II is widely believed, and Maurer [Mau94] showed a reduction of DL to CDH for a wide variety of groups. But for an elliptic curve with $|q| = 170$,

the result in [Mau94] implies at least a 2^{22} loose factor in the reduction². Despite this, we *expect* that CDH has security level essentially 2^{80} , while the reduction argument only enables us to claim that CDH has security level at least 2^{58} . This degradation factor that is not taken into account here is very close to the 2^{30} factor that arises in the CDH1 case. And Assumption I is plausible but far from being deeply studied, since in the case of bilinear group pair pairs we are significantly restricting the set of elliptic curves used. So from our point of view, Conjecture 6 is at least as plausible as considering that assumptions I and II hold.

And even if one has concerns about accepting this conjecture, this new problem captures what means to use a certain scheme with smaller lengths than those provided by a rigorous exact security analysis. In some sense, if a weakness were to be found against such a construction, it should be related to the validity of Conjecture 6. We think this practice is better than using short key sizes ignoring whether the reduction is efficient at all.

5 Boneh-Franklin identity based encryption scheme

In this section we study the identity based encryption (IBE) scheme by Boneh and Franklin [BF01,BF03]. This was the first fully functional IBE scheme designed since the proposal of this idea by Shamir [Sha85], and it was obtained using bilinear maps. The authors introduced the notions of ciphertext indistinguishability against passive and active adversaries for IBE schemes, which are shortly named as IND-ID-CPA and IND-ID-CCA respectively. In our study of this scheme we will strongly rely on [BF01], so we refer the reader there for most of the definitions and security proofs. We cannot directly use the original description of the BF scheme, because it uses bilinear group pairs where $\mathbb{G}_1 = \mathbb{G}_2$, so we must adapt the scheme to the more general case $\mathbb{G}_1 \neq \mathbb{G}_2$. The BF scheme is presented in two stages, starting with a scheme called `BasicIdent`³.

² $\mathcal{O}(\log^3 q)$ calls to a CDH oracle are needed to solve DL using the reduction by Maurer.

³ We choose to minimize the length of the ciphertexts. Therefore we use \mathbb{G}_2 as the set of private keys and then ciphertexts are elements in $\mathbb{G}_1^* \times \{0, 1\}^n$.

BasicIdent

Setup. Let $(\mathbb{G}_1, \mathbb{G}_2)$ a bilinear group pair, with bilinear map $\hat{t} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ and isomorphism ψ . Choose a generator $P_2 \leftarrow \mathbb{G}_2$ and set $P_1 = \psi(P_2)$ (this element is a generator of \mathbb{G}_1). Next pick $s \leftarrow \mathbb{Z}_p^*$ and set $P_{pub} = sP_1 \in \mathbb{G}_1^*$. Choose two cryptographic hash functions

$$H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_2^* \quad \text{and} \quad H_2 : \mathbb{G}_T \rightarrow \{0, 1\}^n.$$

The message space is $\mathcal{M} = \{0, 1\}^n$ and the ciphertext space is $\mathcal{C} = \mathbb{G}_1^* \times \{0, 1\}^n$. The system parameters are

$$\text{params} = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{t}, P_1, P_2, P_{pub}, H_1, H_2 \rangle.$$

The master-key is $s \in \mathbb{Z}_p^*$.

Extract. For a given string $\text{ID} \in \{0, 1\}^*$, compute $Q_{\text{ID}} = H_1(\text{ID}) \in \mathbb{G}_2^*$ and set the private key d_{ID} to be $d_{\text{ID}} = sQ_{\text{ID}} \in \mathbb{G}_2^*$ where s is the master key.

Encrypt. To encrypt $M \in \{0, 1\}^n$ under the public key ID , compute $Q_{\text{ID}} = H_1(\text{ID}) \in \mathbb{G}_2^*$, choose $r \leftarrow \mathbb{Z}_p^*$ and set the ciphertext to be

$$C = \langle rP_1, M \oplus H_2(g_{\text{ID}}^r) \rangle \quad \text{where} \quad g_{\text{ID}} = \hat{t}(P_{pub}, Q_{\text{ID}}) \in \mathbb{G}_T$$

Decrypt. Let $C = \langle U, V \rangle \in \mathcal{C}$ be a ciphertext under the public key ID . To decrypt C using the private key $d_{\text{ID}} \in \mathbb{G}_2^*$, compute $V \oplus H_2(\hat{t}(U, d_{\text{ID}})) = M$.

This completes the description of BasicIdent. The decryption algorithm is sound if and only if $\hat{t}(U, d_{\text{ID}}) = g_{\text{ID}}^r$, and this holds if everything is computed as above, since

$$\hat{t}(U, d_{\text{ID}}) = \hat{t}(rP, sQ_{\text{ID}}) = \hat{t}(P, Q_{\text{ID}})^{sr} = \hat{t}(P_{pub}, Q_{\text{ID}})^r = g_{\text{ID}}^r.$$

In [BF03] it is proven that the above scheme is IND-ID-CPA secure under the BDH assumption in the Random Oracle model. Then an IND-ID-CCA secure scheme is obtained by applying the technique by Fujisaki and Okamoto [FO99b], which transforms a one-way encryption scheme into an IND-CCA encryption scheme in the ROM (we refer to [BDPR98] for public key encryption security notions). If we denote by $E_{\text{pk}}(M, r)$ the encryption of M using the random bits r under the public key pk , the transformation by Fujisaki and Okamoto is the hybrid scheme⁴

$$E_{\text{pk}}^{\text{hy}}(M) = \langle E_{\text{pk}}(\sigma, H_3(\sigma, M)), H_4(\sigma) \oplus M \rangle \quad (1)$$

where σ is random and H_3, H_4 are random oracles. In the case of the BasicIdent scheme, the new hash functions are $H_3 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{Z}_p^*$, $H_4 : \{0, 1\}^n \rightarrow$

⁴ In the case where the symmetric encryption scheme is the one-time pad.

$\{0, 1\}^n$ and $\sigma \in \{0, 1\}^n$. Here follows the description of the new scheme:

FullIdent

Setup. The same as in **BasicIdent** but generating the hash functions H_3, H_4 and adding them to **params**. The message space is $\mathcal{M} = \{0, 1\}^n$ and the ciphertext space is $\mathcal{C} = \mathbb{G}_1^* \times \{0, 1\}^n \times \{0, 1\}^n$.

Extract. Identical to **BasicIdent**.

Encrypt. To encrypt $M \in \{0, 1\}^n$ under the public key **ID**, compute $Q_{\text{ID}} = H_1(\text{ID}) \in \mathbb{G}_2^*$, choose a random $\sigma \in \{0, 1\}^n$, sets $r = H_3(\sigma, M) \in \mathbb{Z}_p^*$ and set the ciphertext to be

$$C = \langle rP_1, \sigma \oplus H_2(g_{\text{ID}}^r), M \oplus H_4(\sigma) \rangle \quad \text{where} \quad g_{\text{ID}} = \hat{t}(P_{\text{pub}}, Q_{\text{ID}}) \in \mathbb{G}_T$$

Decrypt. Let $C = \langle U, V, W \rangle \in \mathcal{C}$ be a ciphertext under the public key **ID**. To decrypt C using the private key $d_{\text{ID}} \in \mathbb{G}_2^*$ do:

1. Compute $V \oplus H_2(\hat{t}(U, d_{\text{ID}})) = M$.
2. Compute $W \oplus H_4(\sigma) = M$.
3. Set $r = H_3(\sigma, M)$. Check that $U = rP$. If not, reject the ciphertext.
4. Output M .

Two additional schemes are needed in order to exhibit the security proof in [BF03]. These schemes are not IBE schemes but merely public key encryption schemes. They are called **BasicPub** and **BasicPub^{hy}**. Here follows the description of

BasicPub

KeyGen. The procedure is similar to **setup** algorithm in **BasicIdent**, but without H_1 . Choose a generator $P_2 \leftarrow \mathbb{G}_2$ and set $P_1 = \psi(P_2)$. Next pick $s \leftarrow \mathbb{Z}_p^*$ and set $P_{\text{pub}} = sP_1 \in \mathbb{G}_1^*$. Choose a hash function $H_2 : \mathbb{G}_T \rightarrow \{0, 1\}^n$. The message space is $\mathcal{M} = \{0, 1\}^n$ and the ciphertext space is $\mathcal{C} = \mathbb{G}_1^* \times \{0, 1\}^n$. The public key is

$$\text{pk} = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{t}, P_1, P_2, P_{\text{pub}}, Q_{\text{ID}}, H_2 \rangle.$$

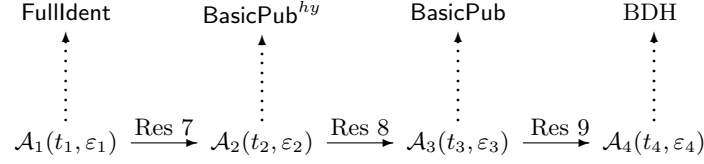
The private key is $\text{sk} = d_{\text{ID}} = sQ_{\text{ID}} \in \mathbb{G}_2^*$.

Encrypt. To encrypt $M \in \{0, 1\}^n$ choose $r \leftarrow \mathbb{Z}_p^*$ and set the ciphertext to be

$$C = \langle rP_1, M \oplus H_2(g^r) \rangle \quad \text{where} \quad g = \hat{t}(P_{\text{pub}}, Q_{\text{ID}}) \in \mathbb{G}_T$$

Decrypt. Let $C = \langle U, V \rangle \in \mathcal{C}$ be a ciphertext under the public key **pk**. To decrypt C using the private key $d_{\text{ID}} \in \mathbb{G}_2^*$, compute $V \oplus H_2(\hat{t}(U, d_{\text{ID}})) = M$.

Finally, the scheme BasicPub^{hy} is the result of applying Fujisaki-Okamoto transformation 1 to the above scheme. The security reduction for FullIdent scheme under the BDH assumption follows the scheme below



The proof of the following results is found in [BF01].

Result 7 Let \mathcal{A}_1 an IND-ID-CCA adversary that has advantage ε_2 against FullIdent making at most q_E private key extraction queries, q_D decryption queries and q_{H_1} hash queries. Then there is an IND-CCA adversary \mathcal{A}_2 that has advantage at least $\frac{\varepsilon_1}{\varepsilon(1+q_E+q_D)}$ against BasicPub^{hy} . Its running time is $t_2 \leq t_1 + c_{\mathbb{G}_1}(q_D + q_H + q_E)$.

Result 8 Let \mathcal{A}_2 an IND-CCA adversary that has advantage ε_2 against BasicPub^{hy} making at most q_D decryption queries and at most q_{H_3}, q_{H_4} hash queries. Then there is an IND-CPA adversary \mathcal{A}_3 that has advantage at least $\frac{1}{2(q_{H_3}+q_{H_4})}[(\varepsilon_2 + 1)(1 - 2/p)^{q_D} - 1]$ against BasicPub . Its running time is $t_3 \leq t_2 + \mathcal{O}((q_{H_3} + q_{H_4}) \cdot (n + \log p))$.

Result 9 Let \mathcal{A}_3 an IND-CPA adversary that has advantage ε_3 against BasicPub making at most q_{H_2} hash queries. Then there is an algorithm \mathcal{B} breaking the BDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ with advantage at least $\frac{2\varepsilon_3}{q_{H_2}}$ and running time $t' \approx t_3$.

Finally, taking into account these reductions, we obtain that BF scheme is $(t_1, q_H, q_D, \varepsilon_1)$ IND-ID-CCA secure if the BDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is

$$\left(t_1 + c_{\mathbb{G}_1}(2q_D + q_H) + 2q_H(n + \log p), \frac{\varepsilon_1}{8eq_H^2q_D} \right)\text{-secure.} \quad (2)$$

The last expression has been simplified replacing any of the hash queries q_{H_i} by q_H and setting $q_D = q_E$. This can be done in this context since we are interested in the tightness of the reduction. Therefore, the security reduction is far from tight. Indeed, there is a $q_H^2q_D$ factor relating the advantages against the scheme and the underlying problem.

Let us compute now the security parameter related to this security reduction for the 2^{80} security level. By property 5, it is assumed that the computational complexity of the BDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is similar to the complexity of DL problem in \mathbb{G}_1 . Then, setting $n = 10^6$ (for instance, think of e-mail applications of IBE schemes) and following the techniques in Section 4.1 we obtain that the expected time to solve BDH using the attacker against the scheme is

$$\frac{t'}{\varepsilon'} \approx e \left(2^{233} + 2^{213} \cdot \frac{\log^3 p}{24 \cdot 500} + 2^{184} \cdot (10^6 + \log p) \right)$$

This security reduction is meaningful when

$$\frac{t'}{\varepsilon'} \leq \left(\frac{t}{\varepsilon} \right)_{\mathbb{G}_1} \quad \text{and} \quad \frac{t'}{\varepsilon'} \leq \left(\frac{t}{\varepsilon} \right)_{\mathbb{G}_T} \quad \text{or} \quad \left(\frac{t}{\varepsilon} \right)_{\mathbb{G}_T}^{\text{LV}}.$$

When computing the key size for a 2^{80} security level, the corresponding calculations leads to $|p| \geq 454$ and $|q| \geq 1980$ ($|q| \geq 2176$ with the LV criterion) for ordinary curves with $\alpha = 6$, and $|q| \geq 1182$ ($|q| \geq 1298$ with the LV criterion) for ordinary curves with $\alpha = 10$. Obviously these values are very large and the protocol is not as practical as with the values proposed in [BF03]. There supersingular curves with $\alpha = 2$ are used to implement the scheme, and the security parameters suggested do not take into account the concrete security. We can compare the computational cost of the scheme with the new parameters to the time required El Gamal in \mathbb{F}_p^* . Regarding encryption, BF scheme would be roughly *400 times slower* than El Gamal encryption! We emphasize again that these results are obtained rigorously following exact security techniques. Then it is very important to obtain tighter security reductions if we want efficient IBE protocols inside the exact security approach.

6 A new identity based encryption scheme with improved tightness

In this section we design a new IBE scheme using the scheme BasicIdent from the previous section and a second general transformation also due to Fujisaki and Okamoto [FO99a]. This conversion uses an IND-CPA encryption scheme and builds an IND-CCA scheme in the ROM. If we denote by $E_{\text{pk}}(M, r)$ the encryption of M using the random bits r under the public key pk , with set of messages $\mathcal{M} = \{0, 1\}^n$, set of coins \mathcal{R} and set of ciphertexts \mathcal{C} , the new transformation is the scheme

$$E_{\text{pk}}^{\text{hyNew}}(M) = E_{\text{pk}}(M||r, H(M||r)) \quad (3)$$

where $M||r \in \{0, 1\}^{n-k_0} \times \{0, 1\}^{k_0}$ and $H : \{0, 1\}^* \rightarrow \mathcal{R}$ is a hash function. Then, $\mathcal{M}^{\text{hyNew}} = \{0, 1\}^{n-k_0}$, $\mathcal{R}^{\text{hyNew}} = \{0, 1\}^{k_0}$ and $\mathcal{C}^{\text{hyNew}} = \mathcal{C}$.

Let us describe the new IBE scheme thereby obtained.

NewFullIdent

Setup. The same as in BasicIdent in Section 5 plus a new hash function $H_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$. **params** are updated with the new changes. The message space is $\mathcal{M} = \{0, 1\}^{n-k_0}$ and the ciphertext space is $\mathcal{C} = \mathbb{G}_1^* \times \{0, 1\}^n$.

Extract. For a given string $ID \in \{0, 1\}^*$, compute $Q_{ID} = H_1(ID) \in \mathbb{G}_2^*$ and set the private key d_{ID} to be $d_{ID} = sQ_{ID} \in \mathbb{G}_2^*$ where s is the master key.

Encrypt. To encrypt $M \in \{0, 1\}^{n-k_0}$ under the public key ID , compute $Q_{ID} = H_1(ID) \in \mathbb{G}_2^*$, choose a random $\sigma \in \{0, 1\}^{k_0}$, sets $r = H_3(M, \sigma) \in \mathbb{Z}_p^*$ and set the ciphertext to be

$$C = \langle rP_1, (M||\sigma) \oplus H_2(g_{ID}^r) \rangle \quad \text{where} \quad g_{ID} = \hat{t}(P_{pub}, Q_{ID}) \in \mathbb{G}_T$$

Decrypt. Let $C = \langle U, V \rangle \in \mathcal{C}$ be a ciphertext under the public key ID . To decrypt C using the private key $d_{ID} \in \mathbb{G}_2^*$ do:

1. Compute $V \oplus H_2(\hat{t}(U, d_{ID})) = M||\sigma$.
2. Sparse $M||\sigma$ and compute $r = H_3(M, \sigma)$. Check that $U = rP$. If not, reject the ciphertext.
4. Output M .

Compared to FullIdent scheme, which is the result of applying FO transformation 1 to BasicIdent, the NewFullIdent scheme presents several advantages:

- I. It provides more compact ciphertexts. In fact, only adds the encoding of an element in \mathbb{G}_1 to the length of the message encrypted.
- II. It presents a tight security reduction to the underlying scheme.
- III. The randomness is a bit string in $\{0, 1\}^{k_0}$, where $k_0 \ll n$ for typical applications⁵.

On the basis of the proof sketched in the previous section, we define in a similar fashion a public key encryption scheme NewBasicPub^{hy} , which is obtained applying conversion 3 to BasicIdent. Then the following results hold:

Result 10 *Let \mathcal{A}_1 an IND-ID-CCA adversary that has advantage ε_2 against NewFullIdent making at most q_E private key extraction queries, q_D decryption queries and q_{H_1} hash queries. Then there is an IND-CCA adversary \mathcal{A}_2 that has advantage at least $\frac{\varepsilon_1}{e(1+q_E+q_D)}$ against BasicPub^{hy} . Its running time is $t_2 \leq t_1 + c_{G_1}(q_D + q_H + q_E)$.*

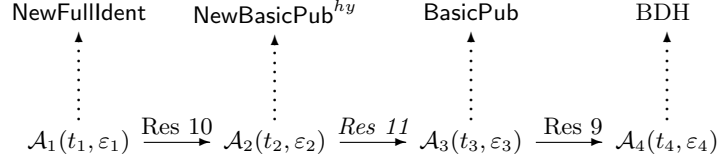
Proof: It is obtained adding minor changes to the proof of Result 7 in [BF01]. It will appear in the final version of these notes.

Result 11 *Let \mathcal{A}_2 an IND-CCA adversary that has advantage ε_2 against NewBasicPub^{hy} making at most q_D decryption queries and at most q_{H_3} hash queries. Then there is an IND-CPA adversary \mathcal{A}_3 that has advantage $(\varepsilon_2 - q_H \cdot 2^{-(k_0-1)})(1 - 1/p)^{q_D}$*

⁵ It lacks to compute the value k_0 .

against **BasicPub**. Its running time is $t_3 \leq t_2 + q_{H_3}(T_{\text{BasicPub}} + \log p)$, where T_{BasicPub} is the running time of **encrypt** in **BasicPub**.

Proof: It is obtained particularizing Theorem 5.4 in [FO00] to our case. A more detailed analysis will appear in the final version of these notes.



First improvement

Finally, taking into account these new reductions, we obtain that **NewFullIdent** scheme is $(t_1, q_H, q_D, \varepsilon_1)$ IND-ID-CCA secure if the BDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is

$$\left(t_1 + c_{\mathbb{G}_1}(2q_D + q_H) + q_H \left(\frac{\log^3 q}{24 \cdot 500} + \log q \right), \frac{\varepsilon_1}{e q_H q_D} \right)\text{-secure} \quad (4)$$

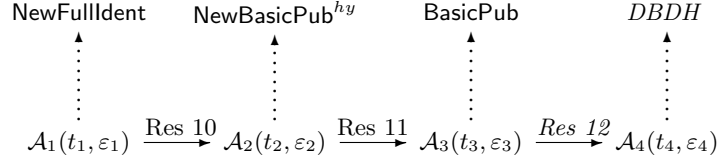
The last expression has been simplified replacing any of the hash queries q_{H_i} by q_H and setting $q_D = q_E$. Then, we get rid of a q_H factor in the BDH advantage with respect to the reduction 2. This has a remarkable effect in computing the security parameter. Now, setting again $n = 10^6$, we obtain the expected time to solve BDH using the attacker against the scheme is

$$\frac{t'}{\varepsilon'} \approx e \left[2^{170} + 2^{150} \cdot \left(\frac{\log^3 p}{24 \cdot 500} + \log q \right) \right]$$

Now, the corresponding calculations lead to $|p| \geq 330$ and $|q| \geq 945$ ($|q| \geq 1069$ with the LV criterion) for ordinary curves with $\alpha = 6$, and $|q| \geq 566$ ($|q| \geq 640$ with the LV criterion) for ordinary curves with $\alpha = 10$. With this new reduction, **NewFullIdent** scheme is roughly 60 times slower in encryption than standard El Gamal. Whereas this is a good improvement with respect to **FullIdent** scheme, it is still not enough.

We can obtain a second tightness improvement using a stronger assumption, namely, the DBDH problem. In this case, we have the following result:

Result 12 *Let \mathcal{A}_3 an IND-CPA adversary that has advantage ε_3 against **BasicPub** making at most q_{H_2} hash queries. Then there is an algorithm \mathcal{B} breaking the DBDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ with advantage roughly ε_3 and running time $t' \approx t_3$.*



Second improvement

With this second tightness improvement, we obtain that **NewFullIdent** scheme is $(t_1, q_H, q_D, \varepsilon_1)$ IND-ID-CCA secure if the DBDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is

$$\left(t_1 + c_{\mathbb{G}_1}(2q_D + q_H) + q_H \left(\frac{\log^3 q}{24 \cdot 500} + \log q \right), \frac{\varepsilon_1}{eq_D} \right)\text{-secure} \quad (5)$$

Then, we get rid of a q_H factor in the security reduction at the cost of relying on a stronger assumption. Let us compute now the security parameter related to this security reduction for the 2^{80} security level. By property 5, it is assumed that the computational complexity of the DBDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is similar to the complexity of BDH problem in this group pair, and in turn to the DL in \mathbb{G}_1 . Then, we obtain the expected time to solve DBDH using the attacker against the scheme is

$$\frac{t'}{\varepsilon'} \approx e \left[2^{110} + 2^{90} \cdot \left(\frac{\log^3 p}{24 \cdot 500} + \log q \right) \right]$$

Now, the corresponding calculations lead to $|p| \geq 213$ and $|q| \geq 356$ ($|q| \geq 426$ with the LV criterion) for ordinary curves with $\alpha = 6$, and $|q| \geq 214$ ($|q| \geq 256$ with the LV criterion) for ordinary curves with $\alpha = 10$. With this new reduction, **NewFullIdent** scheme is roughly 6 times slower in encryption than standard El Gamal, and then we obtain a practical IBE scheme inside the practice-oriented security approach.

Using Remark ? in [KW03], **NewFullIdent** can be modified to present a very tight security reduction to the DBDH problem at the cost of doubling the ciphertext size.

References

- [Bar] P. Barreto. The pairing-based crypto lounge. <http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html> Web page maintained by Paulo Barreto.
- [BB04] D. Boneh and X. Boyen. Short signatures without Random Oracles. In *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73, 2004. An on-line version is available at <http://crypto.stanford.edu/dabo/abstracts/groupsigs.html>.

- [BDPR98] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, 1998.
- [Bel98] M. Bellare. Practice-oriented provable-security. In *1st. International Workshop on Information Security (ISW 97)*, volume 1396 of *Lecture Notes in Computer Science*, pages 221–231, 1998.
- [BF01] D. Boneh and M. Franklin. Identity-Based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO ’01*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, 2001. The full version appears at [BF03].
- [BF03] D. Boneh and M. Franklin. Identity-Based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. This is the full version of [BF01].
- [Bih97] E. Biham. A fast new DES implementation in software. In *Fast Software Encryption 1997*, volume 1267 of *Lecture Notes in Computer Science*, pages 260–272, 1997.
- [BKLS02] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368, 2002.
- [BLS01] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532, 2001. The full version appears at [BLS04b].
- [BLS04a] P.S.L.M. Barreto, B. Lynn, and M. Scott. On the selection of pairing-friendly groups. *Journal of Cryptology*, 17(4):17–25, 2004.
- [BLS04b] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004. This is the full version of [BLS01].
- [BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM CCS*, pages 62–73. ACM Press, 1993.
- [BR96] M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. In *Advances in Cryptology – EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, 1996.
- [Cor00] S. Coron. On the exact security of Full Domain Hash. In *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235, 2000.
- [FO99a] E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In *PKC ’99*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68, 1999. The full version is published in [FO00].
- [FO99b] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology — CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, 1999.
- [FO00] E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. *IEICE Trans. Fundamentals*, E83-9(1):24–32, 2000. This is the full version of [FO99a].
- [FR94] G. Frey and H.G. Rück. A remark concerning m -divisibility and the discrete logarithm problem in the divisor class group of curves. *Mathematics of Computation*, 62:865–874, 1994.
- [JN03] A. Joux and K. Nguyen. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–247, 2003.

- [Jou00] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *ANTS 2000*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394, 2000.
- [KM04] N. Koblitz and A. Menezes. Another look at “provable security”. Cryptology ePrint Archive, Report 2004/152, 2004. <http://eprint.iacr.org/>.
- [KW03] J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In *Proceedings of the 10th ACM CCS*. ACM Press, 2003.
- [LQ04] B. Libert and J.J. Quisquater. The exact security of an identity based signature and its applications. Cryptology ePrint Archive, Report 2004/102, 2004. <http://eprint.iacr.org/>.
- [LV01] A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255–293, 2001.
- [Mau94] U. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In *Advances in Cryptology — CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 271–281, 1994.
- [MOV93] A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Transactions on Information Theory*, 39:1639–1646, 1993.
- [NES03] NESSIE. NESSIE security report. version 2.0, 2003. <http://www.cryptonessie.org/>.
- [Pol78] J.M. Pollard. Monte carlo methods for index computation mod p . *Mathematics of Computation*, 32:918–924, 1978.
- [PSV04] D. Page, N.P. Smart, and F. Vercauteren. A comparison of MNT curves and supersingular curves. Cryptology ePrint Archive, Report 2004/165, 2004. <http://eprint.iacr.org/>.
- [Sch] B. Schneier. <http://www.schneier.com/blowfish-speed.html>.
- [Sha85] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology — CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, 1985.
- [X9.01] ANSI X9.52. Triple data encryption algorithm modes of operation, 2001. American National Standards Institute, 1998.