A Tapestry of Identity-Based Encryption: Practical Frameworks Compared

X. Boyen
Voltage Security, Palo Alto, California, United States.
E-mail: xb@boyen.org

Abstract: This paper surveys the practical benefits and drawbacks of several identity-based encryption schemes based on bilinear pairings. After providing some background on identity-based cryptography, we classify the known constructions into a handful of general approaches. We then describe efficient and fully-secure IBE and IBKEM instantiations of each approach, with reducibility to practice as the main design parameter. Finally, we catalogue the strengths and weaknesses of each construction according to a few theoretical and many applied comparison criteria.

Keywords: Applied identity-based encryption; bilinear pairings; comparison survey.

Reference to this paper should be made as follows: Boyen, X. (2007) 'A Tapestry of Identity-Based Encryption: Practical Frameworks Compared', Int. J. Applied Cryptography, Vol. 1, No. 1, pp.??–??.

Biographical notes: Xavier Boyen obtained his Ph.D. in Computer Science from Stanford University in 2003. He has published extensively in the top cryptology conferences on a range of topics including signatures, encryption, foundations, and human-oriented cryptography. He is known as the inventor of several key algorithms and protocols in pairing-based cryptography. He is currently head scientist at Voltage Security, Inc.

1 INTRODUCTION

Practical constructions of identity-based encryption (IBE) first appeared at the turn of the century. Two similar schemes based on pairings were independently proposed by Sakai et al. (2000) and by Boneh and Franklin (2001), followed soon afterwards by a completely different scheme due to Cocks (2001). These three results in rapid succession came after a comparatively very long *hiatus* of almost two decades, which started when Shamir (1984) first posed his famous IBE question without indicating how to solve it. Although a few interesting attempts at solving the problem have been made in the interval, all pre-2000 proposals were fundamentally inefficient, not only for practical use but even also by the more relaxed standards of complexity theory. We mention the schemes of Tanaka (1987), Tsujii and Itoh (1989), Maurer and Yacobi (1996), and Steiner et al. (1996), for historical value.

Things have changed dramatically in the years since 2000. Many improvements have been made to the Boneh-Franklin IBE scheme, and a few brand new approaches to IBE have even been proposed. All of them, however, rely in one way or another upon the notion of bilinear pairing. Pairings are powerful mathematical constructs defined over certain algebraic curves, and whose recently discovered potential for creative cryptographic applications has not ceased to be a source of much amazement. In this regard, Cocks' pairing-free approach to IBE remains for the most part an isolated result with its share of limitations. Nevertheless, the foregoing is not meant to suggest that all pairing-based IBE systems are mere mirror images of one another; on the contrary, there are significant differences in the way pairings have been used, giving us a choice of systems with dissimilar properties.

In this context, a pairing, also called bilinear map, is a function from two cyclic groups into a third; it is linear in both arguments, and the algebraic groups that define its domain and codomain have computational representations that make the discrete logarithm and a number of related problems infeasibly hard in those groups—or so it is widely believed (a more precise definition will follow). Bilinear pairings have quickly become very prized in cryptography, as much for their demonstrated utility in the construction of new protocols, as for their efficient computability based on the theory of algebraic curves, without which the protocols would not be usable in practice. Identity-based cryptography and IBE in particular can be regarded as the most exemplar success stories of bilinear pairings in cryptography.

In deference to the multiplicity of IBE schemes and the disparity of their properties, we endeavor to draw an inventory of the most practical approaches, and compare the merits and drawbacks of the different frameworks while keeping an eye on usability in actual deployments. This is the objective we pursue in this note. Although other comparisons have been made in the past, our present goal is to adopt a more practically minded viewpoint, intended to serve as a bridge between cryptographers who invent those schemes, and the professional security engineers who must evaluate, select, and eventaully deploy them as part of larger systems.

2 PRELIMINARIES

This section provides general background information for the reader who may not be familiar with the recent literature on identity-based encryption.

2.1 Identity-Based Encryption and Key Encapsulation

The notion of identity-based encryption refers to a special type of public-key encryption where the public key can be freely chosen from an exponentially large set, instead of being generated at random along with the corresponding private key.

Recall that in a traditional public-key cryptosystem, any participant who wishes to be on the receiving end of any encrypted communication must have previously generated a unique key pair. The key pair consists of a secret decryption key, or private key, and a corresponding public encryption key, which must be made widely available in an authenticated manner. Since public keys are just random strings, they are not intrinsically bound to their owner, and so the usual solution to publicize this binding is to employ a combination of public directories and cryptographic certificates issued by a designated trusted authority known as the certificate authority. This often involves a complex architecture referred to as a public-key infrastructure, or PKI.

In identity-based cryptography, a public key can be any string specified externally. These strings can carry their own meaning, such as a combination of the owner's identity and a suitably discretized period of validity. An immediate consequence is that public keys need no longer be distributed nor certified, since they can be reconstructed in full by the encrypting party, and are therefore implicitly trusted. A second consequence is that there must be an efficient mapping from the public key to the corresponding private key; this mapping should of course not be computable by anyone, but only by some trusted authority who holds a master secret for this purpose. In an IBE system, users thus do not compute their own key pairs, but obtain their private keys from the key generation authority, after having shown the adequate credentials or successfully completed a suitable authentication protocol. In IBE contexts, public keys are essentially the same as identities, and often referred as such, or ID for short.

IBE Systems. Abstractly, an identity-based encryption system consists of four cryptographic operations:

Setup(1^σ). A randomized algorithm that takes a unary security parameter σ as input, and outputs a random secret master key *masterk* and the corresponding public system parameters *params*.

Extract$(\mathit{masterk}, \mathsf{ID})$. A deterministic or randomized algorithm that takes as input the master secret *masterk* and a well-formed identity string ID, and outputs a deterministic or randomized private key d_{ID}.

Encrypt$(\mathit{params}, \mathsf{ID}, M)$. A randomized algorithm that takes as input the public parameters *params*, a recipient identity ID, and a message M in a suitable domain, and outputs a ciphertext C.

Decrypt(d_{ID}, C). A usually deterministic algorithm that takes as input a private key d_{ID} and a ciphertext C, and outputs either a decrypted message M or a failure symbol $\perp$.

IBKEM Systems. Sometimes, it is more advantageous to consider the simpler notion of (identity-based) key encapsulation mechanism, which does not allow the sender to choose the message being encrypted. An IBKEM system consists of *Setup*(1^σ) and *Extract*$(\mathit{masterk}, \mathsf{ID})$ as above, plus:

Encapsulate$(\mathit{params}, \mathsf{ID})$. A randomized algorithm that is input the public parameters *params* and a recipient identity ID, and outputs a random session key K and a ciphertext "capsule" C.

Decapsulate(d_{ID}, C). A usually deterministic algorithm that takes as input a private key d_{ID} and a ciphertext capsule C, and outputs either a decrypted session key K or a failure symbol $\perp$.

The primary use of an IBKEM is to encrypt unique random strings to be used as session keys of a downstream Data Encryption Module (DEM) under which the actual message is encrypted.

2.2 Mathematical Background

We now very briefly review the notion of cyclic groups equipped with a bilinear map, or bilinear groups for short. First, a quick reminded of what an algebraic group means in cryptography is in order.

Computational Groups. Let p be a cryptographically large prime, and let $\mathbb{G}$ be a cyclic group of order p. As is common in cryptography, when we say "group" we intend to convey not only the abstract notion of algebraic group, but also the specific computational representation that we chose for it. In our case, although $\mathbb{G}$ is clearly the same algebraic group as $\mathbb{Z}_p^+$ (short for $(\mathbb{Z}/p\mathbb{Z}, +)$), we do not wish to use such a representation for $\mathbb{G}$ because the discrete log is easy in $\mathbb{Z}_p$.

Ideally, the computational structure of $\mathbb{G}$ should give rise to a group representation that is a generic as possible. Groups of points defined on algebraic curves over finite fields (and elliptic curves in particular) are believed to provide such generic-looking representations. This is fortunate because (hyper)elliptic curves are also the one setting in which efficient and cryptographically useful bilinear pairings are known to exist; in particular, all the groups involved in a pairing are believed to sustain the discrete-log hardness assumption.

Bilinear Pairings. Consider a function $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$, where $\mathbb{G}$ and $\hat{\mathbb{G}}$ are cyclic groups of prime order p, and G_t is another group also of order p. Let $g \in \mathbb{G}$ and $\hat{g} \in \hat{\mathbb{G}}$ be generators of their respective groups. The function e will be a bilinear map if it satisfies the following conditions.

Non-degeneracy: $e(g, \hat{g}) \neq 1 \in \mathbb{G}_t$

Bilinearity: $\forall a, b \in \mathbb{Z}, e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$

(Following usual cryptographic conventions, in this paper we employ the multipicative notation for the group operations in $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$, and write 1 for the neutral element.) The groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ are called the bilinear group(s), and $\mathbb{G}_t$ is the target group. Without going into details, we briefly mention how they relate to elliptic curves.

Elliptic Curves. We require the bilinear groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ to have large prime order p. Let E be an elliptic curve over some finite field $\mathbb{F}_q$. We construct $\mathbb{G}$ and $\hat{\mathbb{G}}$ as two linearly independent (sub)groups of order p in the groups of points on E with coordinates in $\mathbb{F}_q$ or in an extension $\mathbb{F}_{q^k}$. It is also possible to take $\mathbb{G} = \hat{\mathbb{G}}$, provided that there be an efficient "distortion" homomorphism to another subgroup $\hat{\mathbb{G}}'$ that is linearly independent of $\mathbb{G}$; the "symmetric" pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ will then be a wrapper for the true pairing computed between $\mathbb{G}$ and $\hat{\mathbb{G}}'$. In all cases, the target group $\mathbb{G}_t$ will be the multiplicative subgroup of order p in the extension field $\mathbb{F}_{q^k}$. The embedding degree k is the smallest extension degree that causes $\mathbb{F}_{q^k}$ to contain a multiplicative subgroup of order p. Since $\mathbb{F}_{q^k}$ is a field, it follows that $\mathbb{G}_t$ will consist of the q-th roots of unity in $\mathbb{F}_{q^k}$. In general, since $\mathbb{G} \prec E(\mathbb{F}_q)$, $\hat{\mathbb{G}} \prec E(\mathbb{F}_{q^k})$, and $\mathbb{G}_t \prec F_{q^k}$, the elements of $\mathbb{G}$ will have much shorter representations than those of $\hat{\mathbb{G}}$ and $\mathbb{G}_t$; however, clever compression tricks do exist.

For these notions to be useful, it is necessary that the pairing and all the group multiplications be computable as efficiently as possible. This constraint, along with the desire to reduce the number of bits needed to represent the various group elements, has motivated several alternative definitions of a pairing, with slight variations designed to accommodate different types of curves. We briefly describe the three main types of pairings, per the nomenclature of Galbraith et al. (2006).

Types of Pairings. The pairing most commonly seen in research papers is the *symmetric* pairing, in which $\mathbb{G} = \hat{\mathbb{G}}$. A symmetric pairing thus reduces to $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$, which has the advantage of simplifying the notation, but is otherwise overly restrictive. It is preferable to describe a scheme using the *asymmetric* definition of a pairing, as given above, which is of the form $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$. Asymmetric pairings are generally more compact and more efficient.

Galbraith et al. (2006) propose a finer classification, that takes into account the ease with which we can move between $\mathbb{G}$ and $\hat{\mathbb{G}}$. If the isomorphism $\phi : \hat{\mathbb{G}} \to \mathbb{G}$ and its inverse ϕ^{-1} are both efficiently computable, the pairing is of type 1, and it is a simple matter to rewrite it as a symmetric pairing. If only ϕ is efficiently computable (but not ϕ^{-1}), then the pairing is of type 2; it is asymmetric as natively defined in $\mathbb{G} \times \hat{\mathbb{G}}$, though it could conceivably be turned into a symmetric pairing in $\hat{\mathbb{G}} \times \hat{\mathbb{G}}$ if efficiency were no object. If none of the isomorphisms is efficiently computable, then the pairing is forcibly asymmetric, and it is said to be of type 3.

Although $\mathbb{G}$ and $\hat{\mathbb{G}}$ may be the same, it is crucial that $\mathbb{G}_t$ remain distinct, and in particular that once we have landed in $\mathbb{G}_t$, it be infeasible to come back into $\mathbb{G}$ or $\hat{\mathbb{G}}$, because doing so efficiently would violate most of the hardness assumptions that make bilinear pairings useful in cryptography. We review a sample of the relevant complexity assumptions in a later section.

The Weil and Tate Pairings. The first efficient algorithm for computing a class of functions on elliptic curves that includes certain bilinear maps was proposed by Miller in 1984, and eventually published in (Miller, 2004). These bilinear maps are known to mathematicians as the Weil and the Tate pairings. More recently, a number of variations to the definition of the Tate pairing have led to substantial performance gains.

We refer the reader to Blake et al. (1999) and Blake et al. (2005) for an algorithmic compendium on this subject. Lynn (2007), in his doctoral dissertation, also provides an excellent resource on pairings and their efficient computation.

2.3 Complexity Assumptions

We briefly define the complexity assumptions upon which the schemes that we shall describe rely. Since the most efficient implementations all make use of the random-oracle model, we focus on the computational form of the various assumptions (and not on the stronger decisional forms, which are usually required for proving indistinguishability in the standard model).

BDH: Bilinear Diffie-Hellman. The BDH problem for a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ is usually stated as follows (Joux, 2004; Sakai et al., 2000; Boneh and Franklin, 2001):

> Given a tuple $(g, g^a, g^b, g^c) \in \mathbb{G}^4$ as input, output $e(g, g)^{abc} \in \mathbb{G}_t$.

The definition has later been relaxed to the general case of an asymmetric pairing $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ (Boneh and Boyen, 2004a):

> Given $(g, g^a, g^b) \in \mathbb{G}^3$ and $(\hat{g}, \hat{g}^a, \hat{g}^c) \in \hat{\mathbb{G}}^3$ as input, output $e(g, \hat{g})^{abc} \in \mathbb{G}_t$.

The two problem definitions coincide when $g = \hat{g}$ and thus $\mathbb{G} = \hat{\mathbb{G}}$, *i.e.*, in the case of type-1 pairings. The (general) BDH assumption then simply states that it is infeasible to solve a random instance of the (general) BDH problem, with non-negligible probability, in time polynomial in the size of the problem instance description.

Gap-BDH: Gap Bilinear Diffie-Hellman. The Gap-BDH problem is essentially the same as the BDH problem, except for the important difference that, here, the solver is given access to a decisional BDH oracle. In other words, in the Gap-BDH problem, the solver must compute a solution to (a random instance of) BDH, while being allowed to make queries to an oracle that can decide whether proposed solutions to arbitrary instances of BDH are correct.

BDHI: Bilinear Diffie-Hellman Inversion. The BDHI problem was originally stated for a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ (Mitsunari et al., 2002; Boneh and Boyen, 2004a), depending on a special additional parameter q:

> Given a $(q + 1)$-tuple $(g, g^x, g^{(x^2)}, \ldots, g^{(x^q)}) \in \mathbb{G}^{q+1}$ as input, output $e(g, g)^{1/x} \in \mathbb{G}_t$.

Again, we use of a more general definition that also works with asymmetric pairings $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$, taken from the full version of (Boneh and Boyen, 2004a):

> Given two tuples $(g, g^x, g^{(x^2)}, \ldots, g^{(x^q)}) \in \mathbb{G}^{q+1}$ and $(\hat{g}, \hat{g}^x, \hat{g}^{(x^2)}, \ldots, \hat{g}^{(x^q)}) \in \mathbb{G}^{q+1}$ as input, output $e(g, \hat{g})^{1/x} \in \mathbb{G}_t$.

The (general) BDHI assumption for some polynomially bounded q states that it is infeasible to solve a random instance of the q-BDHI problem, with non-negligible probability, in time polynomial to the description length of the problem instance.

3 CLASSIFICATION OF IBE FRAMEWORKS

The vast and growing number of identity-based encryption and related schemes that exist today can be traced to one of three known pairing-based approaches to IBE. We characterize them all in this section, using the criteria and terminology from Boyen (2007).

Each framework is characterized by a fundamentally unique way to build an IBE trapdoor from a pairing, and relies on different kinds of assumptions and proofs to realize this core functionality. (Of course, there exist many more than three pairing-based constructions of IBE, but most of them can be traced back to one of the three root paradigms.)

3.1 Full-Domain-Hash IBE

The earliest and, perhaps, conceptually simplest approach to IBE from pairings was discovered independently by Sakai et al. (2000) and by Boneh and Franklin (2001). Even though the scheme of Sakai et al. (2000) was an ID-based key exchange, and that of Boneh and Franklin (2001) an IBE proper, both teams came up with essentially the same ID-based key extraction technique.

The idea is fairly straightforward: it consists of using a cryptographic hash to map an identity string $\mathsf{ID} \in \{0, 1\}^*$ to a bilinear group element $H(\mathsf{ID}) \in \hat{\mathbb{G}}$. With the help of a special public value $g^\alpha \in \mathbb{G}$, any hash value $H(\mathsf{ID})$ can serve as an encryption public key for the identity ID. The corresponding ID-based decryption key is $H(\mathsf{ID})^\alpha$ and can only be computed by the central authority who knows the master key α. (A more detailed description of the Boneh-Franklin system will be given in a later section.)

We propose the name "full-domain hash" for this approach because it crucially relies on hashing the identity directly into a bilinear group, using a hash function which must be modeled as a random oracle. Several schemes fall under this denomination, being direct extensions of the Boneh-Franklin systems; *e.g.*, we mention the hierarchical constructions of Gentry and Silverberg (2002) and Yao et al. (2004). A distinguishing feature of all full-domain-hash IBE systems is that they construct an internal session key of the form $e(g, H(\mathsf{ID}))^{\alpha\, r}$, where α is the master secret and r is an ephemeral encryption randomizer chosen by the sender.

One drawback of this approach is that it makes extensive use of cryptographic hashes (modeled as random oracles), and in particular assumes the availability of hash functions with uniformly distributed images in a bilinear group (*i.e.*, on an elliptic curve). Two problems with this kind of hashing are that it is somewhat expensive, and more importantly that it could potentially restrict the choice of curves since it is not always possible to sample uniformly from or hash evenly into the proper subgroup, without involving the knowledge of any discrete logarithm. Fortunately, the Boneh-Franklin method has been successfully adapted to work with all known types of pairing-friendly curves (sometimes by replacing individual group elements by collective cosets); however, the hashing requirement remains a limitation of the full-domain-hash paradigm that could surface again in the future.

Another drawback of the full-domain-hash framework is that it leads to IBE systems that are significantly less efficient than newer approaches. As we shall see, their inefficiency is due in part because hashing directly on a curve is expensive, and in part because encryption always requires a costly pairing computation that cannot be avoided.

3.2 Exponent-Inversion IBE

The second approach we describe has its roots in a new type of assumption originally proposed by Mitsunari et al. (2002) in the context of a traitor tracing scheme. Their intuition was that any coefficient that appears as the exponent of a group element should be hard to invert, *i.e.*, it should be hard to compute $g^{1/x}$ given g and g^x. However, with the pairing it is easy to "cancel out" the exponent, by pairing g^x with $g^{1/x}$, without having to reveal x itself.

To realize an IBE system from this idea, one would have to encode the identity as part of x, and devise a way to make g^x computable from public information and $g^{1/x}$ from a secret trapdoor. For example, one could define a linear function $x(\mathsf{ID}) = \alpha + \beta\,\mathsf{ID}$, and publish g^α and g^β. A benefit of this "exponent inversion" framework is that we no longer need to hash directly into one of the pairing groups; we merely hash into $\mathbb{Z}_p$ which is an easy operation.

The first scheme based on this approach was proposed by Sakai and Kasahara (2003), without security proof, though a proof was later given by Chen et al. (2005) in the random-oracle model. The first provably secure IBE scheme based on the exponent-inversion principle is the BB_2 system of Boneh and Boyen (2004a), which was designed for the specific purpose of achieving security without random oracles (BB_2 is not to be confused with the completely different BB_1 system from the same paper, discussed next). More recently, Gentry (2006) proposed yet another variation on the exponent-inversion theme, with a tighter security proof than the earlier proposals.

Unfortunately, proving the security of exponent-inversion IBE is not a simple affair. A common characteristic of all these schemes is that their security proofs require surprisingly strong assumptions. A typical assumption for these schemes is q-BDHI, which states that it is hard to compute $e(g, g)^{1/x}$ (and thus $g^{1/x}$) given a polynomially long sequence of elements: g, g^x, g^{x^2}, g^{x^3}, up to g^{x^q}. (Gentry's scheme uses an even stronger assumption which requires specific elements to be expressly omitted from the sequence.) The length of the sequence is determined by a parameter q, which for the known IBE constructions must be at least as large as the maximum number of private key owners that an adversary may corrupt in an active attack (*i.e.*, the number of key extraction that the adversary makes in the IBE security game). Hence, the value of q must be fairly large for the proofs to have any bearing on practice, but then the assumption becomes correspondingly less reassuring.

Indeed, a recent number-theoretic analysis (Cheon, 2006) has shown that these "large-q" assumptions may only provide $O(\sqrt[3]{p})$ concrete security against the generic recovery of x, instead of the larger $O(\sqrt{p})$ concrete security that one would have expected for a generic discrete-log attack. Fortunately, this analysis does not bring into question the credibility of BDHI and similar assumptions, because a matching lower bound $\Omega(\sqrt[3]{p})$ on the generic complexity of breaking BDHI had been previously predicted by Boneh and Boyen (2004b). Nevertheless, the results of Cheon (2006) stress the necessity of correcting for the worst-case value of q (which depends on the threat model) when provisioning a system that relies on a "large-q" assumption.

3.3 Commutative-Blinding IBE

The newest and most appealing IBE framework is that of the BB_1 scheme originally proposed by Boneh and Boyen (2004a). Their method sidesteps most or all the problems associated with the earlier approaches: in particular, the BB_1 scheme allows identities to be encoded (or hashed) as integers as in the exponent-inversion approach, but admits a much better security reduction based on the same BDH complexity assumption as in the full-domain-hash approach (hence, no "large-q" assumption).

Very roughly, the IBE framework of Boneh and Boyen (2004a) is based on the idea of creating, from two or more secret coefficients, two blinding factors that "commute" with each other under the pairing (*i.e.*, the unblinding need not be the reverse of the blinding). The name given to this approach, "commutative blinding", is an attempt to convey the essence of this mechanism.

Perhaps the best quality of the commutative blinding paradigm is the greater flexibility provided by its algebraic structure. Although this approach was the last one to appear, it quickly surpassed the others for the number and variety of extensions to the basic notion of IBE that it has enabled. Here, an "IBE extension" refers to a cryptosystems whose functionality subsumes that of plain IBE at least in some respect. We shall discuss a few examples of extensions in a later section.

3.4 Quadratic-Residuosity IBE (without pairings)

The last IBE approach we characterize is the quadratic residuosity technique used in the Cocks (2001) scheme. It is the only known polynomial-time approach to IBE that does not rely on bilinear pairings. The quadratic residuosity problem is that of determining whether $\exists y : x = y^2 \pmod{N}$ given a composite modulus $N = p_1 p_2$ and a modular residue $x \in \mathbb{Z}_N$. We do not describe the Cocks approach, in part because it does not provide any of the flexibility of the other approachs, and also because each bit of the plaintext is encrypted as one or two residues modulo N, which makes the ciphertexts impractically large even for session keys. The computations are however reasonably fast.

One modification to the Cocks system with a different space/time trade-off was recently proposed by Boneh et al. (2007). Necessarily still in the random-oracle model, the new version enjoys manageable ciphertexts, but suffers from quartic complexity in the security parameter; *i.e.*, encryption time grows as $\Theta(\sigma^4)$ with the number σ of security bits. This is in contrast to the typical cubic complexity of most public-key cryptosystems, including Cocks'.

4 AN ARRAY OF PRACTICAL IBE SYSTEMS

Without further ado, we describe the three main practical IBE constructions that have been proposed in the literature. Each system was chosen as the most efficient representative of one of the three pairing-based IBE paradigms defined in a previous section. For completeness, we consider the IBE as well as an IBKEM version in each of the three frameworks.

Whenever needed, we adapt the published constructions to make use of asymmetric pairings, which are often more efficient than symmetric pairings. We also seek the cheapest way to achieve security against active attacks (chosen-ciphertext and adaptive-identity), which is to use hash functions. Hence, all the schemes we compare are set in the random-oracle model; the rationale is that for practical applications, we would rather rely on the random-oracle heuristic than settle with a less efficient system.

4.1 Boneh and Franklin's BF-IBE

We briefly recall the complete Boneh-Franklin (BF) IBE system, which we also generalize to the setting of asymmetric bilinear maps (if symmetric maps are used, we can assume that $\hat{\mathbb{G}} = \mathbb{G}$). The random-oracle IND-ID-CCA proof of security given by Boneh and Franklin (2001) can be easily generalized to the asymmetric setting under an asymmetric version of the BDH assumption. The bilinear groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ are of prime order p. Identities are represented as bit strings of arbitrary length, and messages are bit strings of some fixed length ℓ. Additionally, we require four cryptographic hash functions viewed as random oracles:

1. a function $H_1 : \{0,1\}^* \to \hat{\mathbb{G}}$ for hashing the recipient identity;

2. a function $H_2 : \mathbb{G}_t \to \{0,1\}^\ell$ for xor-ing with the session key;

3. a function $H_3 : \{0,1\}^\ell \times \{0,1\}^\ell \to \mathbb{Z}_p$ for deriving a blinding coefficient;

4. a function $H_4 : \{0,1\}^\ell \to \{0,1\}^\ell$ for xor-ing with the plaintext.

BF Setup and Key Extraction:

***Setup*:** To generate IBE system parameters, select a random integer $\omega \in \mathbb{Z}_p$, and set $g_{pub} = g^\omega$. The public system parameters *params* and the master secret key *masterk* are given by:

$$params = (g, g_{pub}) \ \in \mathbb{G}^2,$$

$$masterk = \omega \ \in \mathbb{Z}_p.$$

***Extract*:** To generate a private key d_{ID} for an identity $\mathsf{ID} \in \{0,1\}^*$, using the master key ω, the trusted authority computes $h_{\mathsf{ID}} = H_1(\mathsf{ID})$ in $\hat{\mathbb{G}}$ and raises it to the power of the master key. The private key is thus:

$$d_{\mathsf{ID}} = (h_{\mathsf{ID}})^\omega \ \in \mathbb{G}.$$

Full encryption version (Boneh and Franklin, 2001):

***Encrypt*:** To encrypt a message $M \in \{0,1\}^\ell$ for a recipient of identity $\mathsf{ID} \in \{0,1\}^*$, the sender picks a random $s \in \{0,1\}^\ell$, derives $r = H_3(s, M)$, computes $h_{\mathsf{ID}} = H_1(\mathsf{ID})$ and $y_{\mathsf{ID}} = e(g_{pub}, h_{\mathsf{ID}})$, and outputs:

$$C = \Big(g^r, \quad s \oplus H_2({y_{\mathsf{ID}}}^r), \quad M \oplus H_4(s)\Big) \in \mathbb{G} \times \{0,1\}^\ell \times \{0,1\}^\ell.$$

Efficiency may be improved by computing ${y_{\mathsf{ID}}}^r$ as $e({g_{pub}}^r, h_{\mathsf{ID}})$ instead of $e(g_{pub}, h_{\mathsf{ID}})^r$.

***Decrypt*:** To decrypt a given ciphertext $C = (u, v, w)$ using the private key d_{ID}, the recipient successively computes:

$$s = v \oplus H_2(e(u, d_{\mathsf{ID}})),$$

$$M = w \oplus H_4(s),$$

$$r = H_3(s, M).$$

The recipient then verifies that $g^r = u$, and rejects the ciphertext if the equality is not satisfied. Otherwise, the value $M \in \{0,1\}^\ell$ is accepted as the decryption of C.

We have chosen to cast the IBE ciphertext component u in $\mathbb{G}$ and the private key d_{ID} in $\hat{\mathbb{G}}$, under the assumption that elements of $\mathbb{G}$ have a shorter representation than those of $\hat{\mathbb{G}}$. The reverse is also possible, and so the groups may be swapped if hashing proves easier into $\mathbb{G}$ than $\hat{\mathbb{G}}$.

Key encapsulation version; no redundancy (Libert and Quisquater, 2005, adapted):

***Encapsulate*:** To generate a random session key K and encapsulate it for a recipient of identity $\mathsf{ID} \in \{0,1\}^*$, the sender picks a random $r \in \mathbb{Z}_p$, and outputs:

$$K = H_2(e({g_{pub}}^r, H_1(\mathsf{ID}))) \ \in \{0,1\}^\ell,$$

$$C = g^r \ \in \mathbb{G}.$$

***Decapsulate*:** To decrypt a given key encapsulation C using the private key d_{ID}, the recipient simply computes:

$$K = H_2(e(C, d_{\mathsf{ID}})).$$

The preceding IBKEM is a compact variant of Boneh-Franklin and was originally proposed by Libert and Quisquater (2005). It is in fact the most compact IBKEM one can build in the Boneh-Franklin framework. It has no ciphertext redundancy, and therefore cannot provide an explicit test for the recipient to recognize or reject a malformed ciphertext. However, it has a proof of chosen-ciphertext security under the Gap-BDH assumption. Intuitively, if C is malformed, the decrypted session key K will be uniformly random and independent of the true key; thus, when the KEM is used with a suitable downstream DEM, the final decrypted output will be either correct or indistinguishable from random.

Definition 1. *Consider an efficiently computable function $\mathcal{G} : (1^\sigma; r) \mapsto \{e, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_t, g, \hat{g}\}$ that, on input a unary security parameter $\sigma \in \mathbb{N}$ and a binary random string $r \in \{0,1\}^\sigma$, outputs the description of a bilinear pairing and its associated groups. We call the (countably infinite) family induced by $\mathcal{G}$ a* bilinear family.

Theorem 1. *(Boneh and Franklin, 2001, adapted)*

The BF-IBE scheme is secure against probabilistic $O(\sigma^k)$-time IND-ID-CCA adversaries, in the random-oracle model, for all fixed k, when it is implemented in a bilinear family that upholds the BDH assumption.

The BF-IBKEM scheme is secure against probabilistic $O(\sigma^k)$-time IND-IDKEM-CCA adversaries, in the random-oracle model, for all fixed k, when it is implemented in a bilinear family that upholds the Gap-BDH assumption.

4.2 Sakai and Kasahara's SK-IBE

Our next constructions are adapted from the original scheme of Sakai and Kasahara (2003), in the exponent-inversion category. We retain the IBE scheme described by Chen and Cheng (2005), and the IBKEM given by Chen et al. (2005). Unlike the above IBKEM, this one provides an explicit redundancy-based rejection mechanism for malformed ciphertexts. The security of both systems is based on the q-BDHI assumption defined earlier, which is a much stronger assumption than BDH.

Identities are arbitrary bit strings in $\{0,1\}^*$, and messages (or session keys) are fixed-length bit strings in $\{0,1\}^\ell$. As before, $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ is an asymmetric pairing, whose cyclic groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ are respectively generated by g and $\hat{g}$. We need four cryptographic hash functions viewed as random oracles:

1. a function $H_1 : \{0,1\}^* \to \mathbb{Z}_p$ for hashing the recipient identity;
2. a function $H_2 : \mathbb{G}_t \to \{0,1\}^\ell$ for xor-ing with the session key;
3. a function $H_3 : \{0,1\}^\ell \times \{0,1\}^\ell \to \mathbb{Z}_p$ for deriving a blinding coefficient;
4. a function $H_4 : \{0,1\}^\ell \to \{0,1\}^\ell$ for xor-ing with the plaintext.

SK Setup and Key Extraction:

***Setup*:** To generate IBE system parameters, select a random integer $\omega \in \mathbb{Z}_p$, define $g_{pub} = g^\omega$, and compute $v_0 = e(g, \hat{g})$. The public *params* and the secret *masterk* are given by:

$$params = (g, g_{pub}, \hat{g}, v_0) \ \in \mathbb{G}^2 \times \hat{\mathbb{G}} \times \mathbb{G}_t,$$

$$masterk = \omega \ \in \mathbb{Z}_p.$$

***Extract*:** To generate a private key d_{ID} for an identity $\mathsf{ID} \in \{0,1\}^*$, using the master key ω, the trusted authority outputs:

$$d_{\mathsf{ID}} = \hat{g}^{\frac{1}{\omega + H_1(\mathsf{ID})}} \ \in \hat{\mathbb{G}}.$$

Full encryption version (Chen and Cheng, 2005):

***Encrypt*:** To encrypt a message $M \in \{0,1\}^\ell$ for a recipient of identity $\mathsf{ID} \in \{0,1\}^*$, the sender picks a random $s \in \{0,1\}^\ell$, sets $r = H_3(s, M)$ and $g_{\mathsf{ID}} = g_{pub} \cdot g^{H_1(\mathsf{ID})}$, and outputs:

$$C = \Big(g_{\mathsf{ID}}{}^r, \ \ s \oplus H_2(v_0^r), \ \ M \oplus H_4(s)\Big) \\ \in \mathbb{G} \times \{0,1\}^\ell \times \{0,1\}^\ell.$$

***Decrypt*:** To decrypt a ciphertext $C = (u, v, w)$ using the private key d_{ID}, the recipient computes:

$$s = v \oplus H_2(e(u, d_{\mathsf{ID}})),$$

$$M = w \oplus H_4(s),$$

$$r = H_3(s, M).$$

The recipient then verifies that $(g_{pub} \cdot g^{H_1(\mathsf{ID})})^r = u$, and rejects the ciphertext if the equality is not satisfied. Otherwise, the value $M \in \{0,1\}^\ell$ is accepted as the decryption of C.

Key encapsulation version; explicit rejection (Chen et al., 2005):

***Encapsulate*:** To encapsulate a random session key K for a recipient of identity $\mathsf{ID} \in \{0,1\}^*$, the sender selects a random $s \in \{0,1\}^\ell$, sets $r = H_3(s, \perp)$ and $g_{\mathsf{ID}} = g_{pub} \cdot g^{H_1(\mathsf{ID})}$, and outputs:

$$K = H_4(s) \ \in \{0,1\}^\ell,$$

$$C = \Big(g_{\mathsf{ID}}{}^r, \ \ s \oplus H_2(v_0^r)\Big) \ \in \mathbb{G} \times \{0,1\}^\ell.$$

Decapsulate: To decrypt a key encapsulation $C = (u, v)$ using the private key d_{ID}, the recipient first computes

$$s = v \oplus H_2(e(u, d_{\mathsf{ID}})),$$

$$r = H_3(s, \perp),$$

then tests whether both

$$(g_{pub} \cdot g^{H_1(\mathsf{ID})})^r = u \quad \text{and} \quad s \oplus H_2(v_0^r) = v.$$

If either equality fails, the ciphertext is rejected. Otherwise, the session key is decrypted as:

$$K = H_4(s).$$

These constructions illustrate that a chosen-ciphertext secure KEM with explicit ciphertext rejection may be just as complex as a full encryption scheme based on the same complexity assumption.

Theorem 2. *(Chen and Cheng, 2005, adapted) (Chen et al., 2005, adapted)*

The SK-IBE scheme is secure against probabilistic $O(\sigma^k)$-time IND-ID-CCA adversaries, in the random-oracle model, for all fixed k, when it is implemented in a bilinear family that upholds the q-BDHI assumption for $q \leq O(\sigma^{k'})$ and some fixed k'.

The SK-IBKEM scheme is secure against probabilistic $O(\sigma^k)$-time IND-IDKEM-CCA adversaries, in the random-oracle model, for all fixed k, when it is implemented in a bilinear family that upholds the q-BDHI assumption for $q \leq O(\sigma^{k'})$ and some fixed k'.

4.3 Boneh and Boyen's BB$_1$-IBE

The last system we describe is an optimized version of the first IBE system proposed by Boneh and Boyen, or BB$_1$. The scheme they originally gave in (2004a) came with a security reduction in the standard model against selective-identity attacks. Since for practical applications the goal is to get the full adaptive-identity, chosen-ciphertext (IND-ID-CCA) security guarantees without sacrificing performance, we shall describe suitably augmented versions of BB$_1$ in the random-oracle model. We first give an IBE scheme with explicit ciphertext validation, followed by a compact IBKEM. We describe both assuming asymmetric pairings; but once again the symmetric-pairing versions can be obtained by setting $\mathbb{G} = \hat{\mathbb{G}}$ and dropping all the "hats" ($\hat{\ }$) from the notation.

Identities are represented using distinct arbitrary bit strings in $\{0,1\}^*$. The messages (or session keys) are bit strings in $\{0,1\}^\ell$ of some fixed length ℓ. As usual, $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ is the pairing, and g and $\hat{g}$ respectively generate the bilinear groups $\mathbb{G}$ and $\hat{\mathbb{G}}$. We require the availability of three cryptographic hash functions viewed as random oracles:

1. a function $H_1 : \{0,1\}^* \to \mathbb{Z}_p$ for hashing the recipient identity;
2. a function $H_2 : \mathbb{G}_t \to \{0,1\}^\ell$ for xor-ing with the cleartext;
3. a function $H_3 : \mathbb{G}_t \times \{0,1\}^\ell \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_p$ to make the ciphertext non-malleable.

BB$_1$ Setup and Key Extraction:

Setup: To generate IBE system parameters, first select three integers α, β, and $\gamma \in \mathbb{Z}_p$ at random. Set $g_1 = g^\alpha$ and $g_3 = g^\gamma$ in $\mathbb{G}$, and compute $v_0 = e(g, \hat{g})^{\alpha\beta}$. (Note that $g_2 = g^\beta$ is not needed.) The public system parameters *params* and the master secret key *masterk* are given by:

$$\mathit{params} = (g, g_1, g_3, v_0) \ \in \mathbb{G}^3 \times \mathbb{G}_t,$$

$$\mathit{masterk} = (\hat{g}, \alpha, \beta, \gamma) \ \in \hat{\mathbb{G}} \times {\mathbb{Z}_p}^3.$$

Strictly speaking, the generator $\hat{g}$ need not be kept secret, but since it will be used exclusively by the authority, it can be retained in *masterk* rather than published in *params*.

Extract: To generate a private key d_{ID} for an identity $\mathsf{ID} \in \{0,1\}^*$, using the master key, the trusted authority picks a random $r \in \mathbb{Z}_p$ and outputs:

$$d_{\mathsf{ID}} = \left(\hat{g}^{\alpha\beta + (\alpha H_1(\mathsf{ID}) + \gamma) r}, \ \ \hat{g}^r\right) \ \in \hat{\mathbb{G}} \times \hat{\mathbb{G}}.$$

Full encryption version (Boneh and Boyen, 2004a):

Encrypt: To encrypt a message $M \in \{0,1\}^\ell$ for a recipient $\mathsf{ID} \in \{0,1\}^*$, the sender first picks a random $s \in \mathbb{Z}_p$, computes $k = v_0^s \in \mathbb{G}_t$, assigns $c = M \oplus H_2(k) \in \{0,1\}^\ell$, calculates $c_0 = g^s$ and $c_1 = g_3^s \, g_1^{H_1(\mathsf{ID})\, s}$ in $\mathbb{G}$, sets $t = s + H_3(k, c, c_0, c_1) \bmod p$, and then outputs:

$$C = \left(c, \ c_0, \ c_1, \ t\right) \ \in \{0,1\}^\ell \times \mathbb{G} \times \mathbb{G} \times \mathbb{Z}_p.$$

Decrypt: To decrypt a given ciphertext $C = (c, c_0, c_1, t)$ using the private key $d_{\mathsf{ID}} = (d_0, d_1)$, the recipient computes:

$$k = e(c_0, d_0)/e(c_1, d_1) \ \in \mathbb{G}_t,$$

$$s = t - H_3(k, c, c_0, c_1) \ \in \mathbb{Z}_p.$$

Then, if the component-wise equality $(k, c_0) \stackrel{?}{=} \left(v_0^s, g^s\right)$ does not hold for both elements, the ciphertext is rejected. Otherwise, the plaintext is given by:

$$M = c \oplus H_2(k) \ \in \{0,1\}^\ell.$$

Key encapsulation version; implicit rejection:

Encapsulate: To generate a random session key and encapsulate it for a recipient with identity $\mathsf{ID} \in \{0,1\}^*$, the sender picks a random $s \in \mathbb{Z}_p$ and outputs:

$$K = H_2(v_0^s) \ \in \{0,1\}^\ell,$$

$$C = \left(g^s, \ \ g_3^s \, g_1^{H_1(\mathsf{ID})\, s}\right) \ \in \mathbb{G} \times \mathbb{G}.$$

The cleartext session key is K; the encapsulated session key is C.

Decapsulate: To decapsulate an encrypted session key $C = (c_0, c_1)$ using the private key $d_{\mathsf{ID}} = (d_0, d_1)$, the recipient outputs:

$$K = H_2\big(e(c_0, d_0)/e(c_1, d_1)\big) \ \in \{0,1\}^\ell.$$

The IBKEM ciphertext contains no redundancy that allows the explicit rejection of a malformed ciphertext; chosen-ciphertext security follows from the fact that incorrectly decrypted session keys are randomly distributed. (We note however that, contrarily to the Boneh-Franklin IBKEM, the encapsulated keys here contain exactly enough redundancy to verify the recipient's identity.)

Theorem 3. *(Boneh and Boyen, 2004a, adapted)*
The BB_1-IBE scheme is secure against probabilistic $O(\sigma^k)$-time IND-ID-CCA adversaries, in the random-oracle model, for all fixed k, when it is implemented in a bilinear family that upholds the BDH assumption.
The BB_1-IBKEM scheme is secure against probabilistic $O(\sigma^k)$-time IND-IDKEM-CCA adversaries, in the random-oracle model, for all fixed k, when it is implemented in a bilinear family that upholds the Gap-BDH assumption.

4.3.1 BB_1 Variant for Verifiable Threshold Applications

The master key in BB_1 can be given in a slightly less efficient form that only consists of group elements in $\hat{\mathbb{G}}$ instead of their exponents in $\mathbb{Z}_p$. The extraction algorithm needs to be adapted slightly, but since it extracts private keys with the same distribution as before, the encryption and decryption algoritms remain unchanged. The modified algorithms are as follows.

Setup′: Like *Setup*, except that we also compute $\hat{g}_1 = \hat{g}^\alpha$, $\hat{g}_3 = \hat{g}^\gamma$ and $\hat{g}_0 = \hat{g}^{\alpha\beta}$ in $\hat{\mathbb{G}}$ (so that $v_0 = e(g, \hat{g}_0) = e(g_1, \hat{g}_2)$ if we let $\hat{g}_2 = \hat{g}^\beta$) and retain the alternate master key as: $\mathit{masterk}' = (\hat{g}, \hat{g}_0, \hat{g}_1, \hat{g}_3) \in \hat{\mathbb{G}}^4$.

Extract′: Pick a random $r \in \mathbb{Z}_p$ and compute ID's private key as: $d_{\mathsf{ID}} = \big(\hat{g}_0\, \hat{g}_3^r\, \hat{g}_1^{H_1(\mathsf{ID})\, r}, \hat{g}^r\big)$.

This variant is beneficial for applications based on Verifiable Secret Sharing (VSS), such as the cryptosystem of Boneh et al. (2006) which provides chosen-ciphertext security with non-interactive threshold decryption. Key extraction in the VSS variant of BB_1 requires three fixed-base exponentiations in $\hat{\mathbb{G}}$ instead of two.

4.3.2 BB_1 Key Encapsulation for Multiple Recipients

Earlier we mentioned that it is usually infeasible or unsafe to try to encapsulate the same session key for multiple recipients in a KEM. Unfortunately, this is a prerequisite for reusing the DEM component in a hybrid ciphertext intended to encrypt the same plaintext for multiple recipients. The solution in such cases is to use an intermediate "thin" layer between the KEM and the DEM. We show how this is done with the BB_1-IBKEM, which requires an additional hash function:

4. a function $H_4 : \{0,1\}^\ell \to \{0,1\}^\ell$ for deriving an intermediate session key.

Encapsulate′: To generate a random session key and encapsulate it for multiple recipients with identities $\mathsf{ID}_i \in \{0,1\}^*$, the sender selects one random string $u \in \{0,1\}^\ell$, picks an ephemeral random $s_i \in \mathbb{Z}_p$ for each recipient, and outputs:

$$K = H_4(u) \ \in \{0,1\}^\ell,$$

$$C_i = \Big(u \oplus H_2(v_0^{s_i}),\ \ g^{s_i},\ \ g_3^{s_i}\, g_1^{H_1(\mathsf{ID}_i)\, s_i}\Big) \in \{0,1\}^\ell \times \mathbb{G} \times \mathbb{G}.$$

The cleartext session key is K; the encapsulated session key given to recipient ID_i is C_i.

Decapsulate′: To decapsulate an encrypted session key $C = (c, c_0, c_1)$ using the private key $d_{\mathsf{ID}} = (d_0, d_1)$, the recipient outputs:

$$K = H_4\Big(c \oplus H_2\big(e(c_0, d_0)/e(c_1, d_1)\big)\Big) \ \in \{0,1\}^\ell.$$

The multi-recipient IBKEM is more complex, and indeed, starts to resemble the full IBE scheme.

4.4 Other IBE Systems

For the sake of completeness, we also mention the following IBE schemes from the literature. All of these systems have theoretical appeal and are reasonably efficient, but none of them appears to be as practical as the IBE systems mentioned above, and so we will not describe or compare them explicitly.

- Cocks' IBE scheme (2001). We already mentioned this system as the only known reasonably efficient approach to IBE that is not based on the theory of bilinear pairings. We omit it from this survey because the pairing-based systems are much more efficient and much better suited for practical applications.

- Boneh and Boyen's BB_1 without random oracles, which is the original version described in (2004a, §4). The "pure" version of BB_1 is less efficient than the random-oracle version described in the present survey. Specifically, the original version has an efficient security proof only in the selective-identity attack model, and requires bilinear groups with more than twice the bit size in order to be secure against full adaptive-identity opponents at the same security level as the random-oracle version; *cf.* Boneh and Boyen (2004a, §7).

- Boneh and Boyen's BB_2 without random oracles (2004a, §5). Although this scheme can also be made secure against active attacks either with larger groups or in the random-oracle model, the resulting scheme would remain dominated in all respect by the corresponding version of BB_1. This is why, for our comparison, we selected the IBE scheme of Sakai and Kasahara instead, which at least appears to have shorter ciphertexts.

- Waters' scheme (2005), and its subsequent improvements due to Chatterjee and Sarkar (2005) and Naccache (2005), all of which belong to the framework of the BB_1 system. They offer tighter proofs of security without random oracles, at the cost of significantly reduced efficiency. Naccache (2005) in particular claims that his scheme is efficient enough for use in practice, but in reality it still suffers from larger public parameters and is less efficient than the schemes already described, especially if chosen-ciphertext security is a requirement.

- Gentry's scheme (2006). This scheme bears a lot of resemblance to BB_2, and has the advantage of a tighter proof of security without random oracles. On the negative side, the Gentry scheme is less efficient than SK and the random-oracle version of BB_2, and requires an even stronger complexity assumption (of the "large q" variety; see Section 2.3). Chosen-ciphertext secure versions of the system are also twice as expensive, unless random oracles are used.

In Search of Practical IBE Without Random Oracles. It was already noted that several IBE systems can be used without random oracles; some of them rather clumsily (Boneh and Boyen's BB_1 and BB_2); others more elegantly (Waters', Chatterjee and Sarkar's, Naccache's, and Gentry's). However, none of them was deemed practical enough to be retained in our comparison.

One should keep in mind that the absence of random oracles makes these schemes more expensive, not exclusively because of the adaptive-identity security requirement, but also when striving for the more mundane notion of chosen-ciphertext security. Indeed, since these two notions of active security are orthogonal, they must be dealt with separately, and each has its own cost.

Most schemes in the theoretical literature are presented in basic semantic security fashion, notwithstanding that active attacks are a real concern in many practical applications. The reason why chosen-ciphertext IBE security without random oracle is expensive is that it can be achieved in one of two ways: double encryption (Naor and Yung, 1990; Cramer and Shoup, 1998), and hiearchical IBE (Canetti et al., 2004). Either way, the core system ends up being duplicated, which results in ciphertexts (and public parameters) that are twice as long and twice as expensive to create, in comparison to the basic semantically secure system which may already be rather costly.

4.5 Full Encryption *vs.* Key Encapsulation

Recall that the main difference between encryption and key encapsulation (and, by extension, between IBE and IBKEM) is that in the former the sender chooses the message, while in the latter the "message" is a random string meant to be used in a downstream symmetric-key data encryption module, or DEM. In practice, both IBE and IBKEM schemes will likely be used to encrypt a short random key, because a symmetric-key DEM will always be far more efficient for long messages. A benefit of using full encryption for the public-key header is to make it easier to encrypt the same session key multiple times under different identities, which is important in applications such as email where the same message is encrypted for multiple recipients. With a KEM header, it would be necessary to interpose a third layer between KEM and DEM.

The main advantage of key encapsulation over full public-key encryption is that it encourages the conceptual modularization of asymmetric- and symmetric-key operations. Another advantage is the possibility of achieving very compact ciphertexts by letting the KEM and the DEM lean on each other for the chosen-ciphertext security the whole system (Shoup, 2001). In particular, it is known how to build suitable DEMs without redundancy from block ciphers using certain modes of operation such as CMC (Halevi and Rogaway, 2003) or EME (Halevi and Rogaway, 2004), and depending on the complexity assumptions one is willing to make, it is possible to eliminate all redundancy from the KEM too (but not all randomness, of course). We saw an example of "KEM without redundancy" with the Boneh-Franklin IBKEM described above.

There are disadvantages to the key encapsulation route, however. We already mentioned the difficulties of encrypting the same message for multiple recipients. Second, and more importantly, the way to transform a semantically secure encryption scheme into a chosen-ciphertext secure KEM without redundancy (in the random-oracle model) is to hash the implicit session key of the encryption scheme; but the security of the resulting construction will then depend on a "Gap" version of whichever complexity assumption was used for the encryption scheme (*e.g.*, Gap-BDH instead of BDH): for assumptions that were already stated in bilinear groups, this constitutes a major leap of faith. Both the BF-IBKEM and BB_1-IBKEM above use this stronger assumption.

An alternative is to build a KEM essentially in the same way as a chosen-ciphertext secure encryption scheme, adding redundancy for the explicit rejection of malformed ciphertexts. A generic KEM construction based on this approach was recently given by Bentahar et al. (2005). A drawback of this approach is that it heightens the complexity of KEM to that of full encryption, and so one might as well prefer the added convenience of encrypting the session key into a self-contained IBE ciphertext. The Sakai-Kasahara IBKEM described above is an example of redundant key encapsulation with explicit ciphertext validation. As we already noted, such "KEMs with redundancy" offer only a small efficiency gain over the corresponding chosen-ciphertext secure encryption scheme with explicit validation.

5 PRACTICAL COMPARISONS

We are now ready to draw factual comparisons between our three frameworks: BF, SK, and BB_1. We shall focus our attention on the various features that one could seek in practical deployments. To ensure unbiased comparisons, we shall endeavor to support our conclusions with specific data for both the IBE and IBKEM representatives of each framework.

Notation. In the sequel, we shall make a number of qualitative Yes/No comparisons; and in order to convey the (un)desirability of the various attributes, we shall use different symbols to signify that an attribute is present. We define these symbols on Table 1.

5.1 Hardness Assumptions and Security Models

Our first comparison criterion is about the security guarantees that one can obtain from a scheme. There are three main concerns:

Security definition: it clearly matters which notion of security is satisfied by a scheme, *i.e.*, whether semantic security holds against chosen-plaintext or chosen-ciphertext attacks, and whether the adversary is confined to a selective-identity attack model (Canetti et al., 2003) or may mount a full adaptive-identity attack (Boneh and Franklin, 2001).

Because the focus of this note is exclusively on practical schemes, we always insist on the strongest security model: indistinguishability against adaptive chosen-ciphertext and adaptive-identity attacks, or IND-ID-CCA; we refer to (Boneh and Franklin, 2001) for a formal definition.

Proof model: we distinguish whether security reductions are made in the standard model or in an idealized model such as the random-oracle model.

Although it would be nice to avoid random oracles, this invariably comes at a performance hit that is unwelcome in practice. In the case of IBE, several efficient schemes have been constructed in the standard model (Boneh and Boyen, 2004a; Waters, 2005; Gentry, 2006); however, the performance gap with the best random-oracle constructions remains significant, especially if we consider instantiations of those schemes that are secure against IND-ID-CCA opponents. (Recall that chosen-ciphertext security in the standard model does not come for free.)

Hardness assumption: it is also important to make note of the particular complexity assumption on which the security of a scheme is based, and to assess the risks that are inherent to it.

Although many complexity assumptions have been proposed in bilinear groups, sometimes without proper justification, only a handful have been used to construct actual IBE systems; they can further be divided into a mild and a not-so-mild category.

The "not-so-mild" assumptions include q-BDHI for large values of the parameter q, *e.g.*, as when q must grow with the number of queries made by a real-life attacker against the system. Because then the problem instances are impractically large, assumptions such as these are not efficiently falsifiable in the sense of Naor (2003).

The "mild" assumptions include Bilinear Diffie-Hellman (Joux, 2004; Joux and Nguyen, 2003) and Linear (Boneh et al., 2004a); these have very succinct problem instances that need not depend on the adversary's capabilities, and are thus efficiently falsifiable.

We remark that, in all cases, indistinguishability reductions in the standard model will typically require the decisional version of an assumption, whereas the computational version often suffices for reductions in the random-oracle model. This is an often overlooked benefit of using hash functions in security proofs. Shoup (2000) and Cramer and Shoup (2002) have proposed security reductions based on hash-decisional complexity assumptions, which are a middle ground between the weaker computational and the stronger decisional complexity assumptions, that sometimes is just strong enough to allow a proof to go through outside of the random-oracle model. Boneh and Boyen (2004a) have also used this approach in the context of identity-based encryption.

Security Comparison

Table 2 shows that all systems satisfy the notion of adaptive chosen-ciphertext security; but the security reductions require differing assumptions.

All systems as described use the random-oracle model for most or all security properties; however the reliance on that model is not equally crucial in all schemes. Without going into details, we make the following general observations:

- "Full domain hash" systems (BF) are highly dependent on random oracles for essentially all of their security properties (chosen-ciphertext security and identity-based collusion resistance). No provable security remains outside of the random-oracle model for these schemes.

- "Exponent inversion" schemes rely somewhat strongly on random oracles, depending on the construction. The SK scheme described here requires the random-oracle model both for collusion resistance and for chosen-ciphertext security.

Table 1: Symbols used to express the desirability of met and unmet ("Yes" and "No") attributes.

"YES" symbols:	
(✓)	a checkmark signals that a feature is present and desirable;
(∼)	a tilde denotes a property that is of ambivalent quality;
(×)	a cross marks a characteristic that is detrimental in nature;
"NO" symbol:	
( )	an empty space indicates the *absence* of a characteristic, whether good or bad.

Table 2: Relationship between security properties and complexity assumptions for the described instances of BF, SK, and BB_1.

	Full IBE			IBKEM Only		
	BF-IBE	SK-IBE	BB_1-IBE	BF-KEM	SK-KEM	BB_1-KEM
Provable Security						
IND-ID-CCA	✓(RO)	✓(RO)	✓(RO)	✓(RO)	✓(RO)	✓(RO)
and without RO?	none	sID-CPA*	sID-CPA*/**	none	none	sID-CPA*/**
Assumption Strength						
q-BDHI (large q)		×			×	
Gap-BDH				CCA***		CCA***
BDH	✓		✓	CPA***		CPA***

* Adaptive-ID security does hold if the group order is much larger than the size of the identity space (Boneh and Boyen, 2004a, §7).

** Converting CPA into CCA security is possible by extending the scheme into a 2-hierarchy (Boneh and Boyen, 2004a, §6).

*** CCA security needs Gap-BDH due to lack of ciphertext redundancy; CPA security needs only computational BDH.

The previously mentioned BB_2 and Gentry schemes fare better in that respect. BB_2 remains collusion-resistant without random oracles if we either relax the attack model or increase the bilinear group size; BB_2 also retains full chosen-ciphertext security via a recent hierarchical construction due to Boyen (2007). The closely related system of Gentry (2006) achieves full IBE security directly and efficiently, without using random oracles.

- "Commutative blinding" systems (BB_1) are the most amenable to random oracle elimination. For instance, BB_1 can make good use of random oracles to enhance its security, but remains secure in the standard model under a weaker notion of security, or if we increase the bilinear group size; see Boneh and Boyen (2004a, §7) for details. Alternatively, BB_1 can be modified to get equivalent security without random oracles without excessive degradation of performance; see Waters (2005) and Naccache (2005).

5.2 Pairing Compatibility

The second criterion we need to consider is how demanding the scheme is on the pairing function. We have already seen that pairings can be of different kinds: type-1 pairings are symmetric; type-2 pairings are asymmetric but support a one-way mapping from $\hat{\mathbb{G}}$ to $\mathbb{G}$; type-3 pairings are asymmetric and the bilinear groups cannot be mapped efficiently to each other. Some schemes will not work without an efficient uni- or bi-directional mapping; others will be insecure if an efficient mapping is available. Yet other schemes are completely indifferent to either the presence of the absence of homomorphisms, and will let us allocate the cryptographic data between $\mathbb{G}$ and $\hat{\mathbb{G}}$ in the optimal way.

Additionally, some hash functions that are modeled as random oracles may be difficult to instantiate, because some but not all curve types will let us map directly into the bilinear groups. For the known pairing constructions, type-1 and type-3 curves always support hashing, but for type-2 curves, direct hashing only works into the codomain of the efficiently computable isomorphism (*i.e.*, into $\mathbb{G}$ if the isomorphism goes from $\hat{\mathbb{G}}$ to $\mathbb{G}$). This could be a problem with IBE schemes that require direct hashing into the bilinear group $\hat{\mathbb{G}}$. (Hashing into $\mathbb{Z}_p$ is always easy regardless of the choice of curve.)

Compatibility Comparison

Table 3 shows the types of pairings that can be used to implement each system. To summarize,

Table 3: Pairing and bilinear group requirements for the described instances of BF, SK, and BB_1.

	Full IBE			IBKEM Only		
	BF-IBE	SK-IBE	BB_1-IBE	BF-KEM	SK-KEM	BB_1-KEM
Pairing Compatibility						
type 1	✓	✓	✓	✓	✓	✓
type 2	( )*	✓	✓	( )*	✓	✓
type 3	✓	✓	✓	✓	✓	✓
Need Homomorphism?						
for proof		×			×	
for scheme						
Hash into Group?						
into $\mathbb{G}$						
into $\hat{\mathbb{G}}$	×			×		
into $\mathbb{G}_t$						

* Type-2 and type-3 pairings are implemented using different groups on the same curves; type-3 allow hashing into $\hat{\mathbb{G}}$.

- "Full domain hash" systems (BF) require the ability to hash directly into the bilinear group $\hat{\mathbb{G}}$, but do not require an efficient homomorphism between $\mathbb{G}$ and $\hat{\mathbb{G}}$.
- "Exponent inversion" schemes (SK) rely on the existence of an efficient homomorphism for the security reduction, though not for the scheme itself. Refinements of the proof technique might let us dispense with this requirement altogether; see for example the full version of (Boneh and Boyen, 2004b).
- "Commutative blinding" systems (BB_1) place the weakest demands on the bilinear groups, requiring neither hashing nor the presence or absence of an efficiently computable homomorphism.

5.3 Intrinsic Versatility

The third criterion we consider is not as easy to circumscribe precisely. It addresses the question of how easily a given scheme can be adapted to solve a practical requirement. Generally, the more features a scheme supports, or is natively compatible with, the more likely it is that it will be useful for an intended application.

We first mention a couple of features that are very dependent upon a particular scheme's construction:

Multi-recipient encryption: It is desirable when encrypting the same (long) message for multiple recipients, to have a single common encryption body with a different header for each recipient. This is very commonly done using a single symmetric-key body and multiple asymmetric-key headers, using hybrid encryption.

Streaming (for chosen-ciphertext security): Often, it is a requirement that encryption and/or decryption be performed on-the-fly without buffering, *i.e.*, in a streaming manner. This is particularly important for very long plaintexts, or when the size of the data stream is not known in advance.

In principle, streaming encryption is always possible (though not all schemes support it). However, the very notion of unbuffered decryption clashes with chosen-ciphertext security, since the former requires that the decrypted plaintext be streamed to the output before the end of the ciphertext stream has been reached, which is forbidded by CCA security. Nevertheless, it is useful to consider a special-purpose notion of decryption streaming, which presupposes that the consumer application can be trusted not to exploit a partially decrypted stream if it later turns out that the ciphertext must be rejected.

Thus, we define the notion of "CCA-secure streaming" as a restricted notion of chosen-ciphertext security where the decryption algorithm is allowed to start outputting a decryption stream on an invalid ciphertext, provided that it indicates at the end of the stream whether the entire plaintext should be accepted or rejected. This is idealized in the security model by withholding the decrypted stream from the adversary's view until the decryption algorithm has given its final accord.

In addition to the above construction-specific properties, we also consider certain useful extensions to the basic notion of IBE: these are more complex features that are typically not satisfied natively by the basic schemes, but may added at little cost if the underlying framework supports it. These features are as follows:

Threshold secret sharing: This is the ability to divide the master secret into n shares given to separate authorities, in order to avoid the concentration of power into a single entity. In threshold IBE, authorities can only create private key shares that by themselves are useless for decryption. A user has to obtain t valid shares from t different authorities in order to assemble his or her private key.

Hierarchical identities: This refers to the arrangement of identities into a (public, syntactic) hierarchy, in such a way that any member of the hierarchy can act as a local authority for all the subordinate identities. The notion of HIBE, or Hiearchical IBE, was first defined by Horwitz and Lynn (2002), and first achieved by Gentry and Silverberg (2002).

Forward security: In the context of IBE, forward security refers to the authority's ability to evolve the master key forward in time, according to a well-defined discrete schedule with a certain time granularity. The goal is to prevent today's master key (at some time T) to be used to decrypt yesterday's ciphertexts (whether directly, or indirectly by extracting a suitable decryption key).

Because the number of periods is potentially very large, depending on the time granularity, it is important that the efficiency of the scheme not be too sensitive this number: ideally, the dependence should be logarithmic at most. See Canetti et al. (2003), Yao et al. (2004), and Boneh et al. (2005) for successive refinements on IBE forward security.

Versatility Comparison

Table 4 lists certain important extensions that may or may not be implemented in one or the other approach. The "commutative blinding" BB_1 schemes are the most versatile, closely followed by the "full domain hash" BF schemes. Until very recently, it was believed that the "exponent inversion" schemes such as SK were severely limited expansion-wise, but recent work (Boyen, 2007) has shown the latter approach to be more flexible than previously thought. Table 4 thus shows the known capabilities of the three main frameworks as of today.

5.4 Time and Space Efficiency

The most immediate comparison criterion is of course the efficiency of a scheme. This includes the computation time of the most common operations, and the size of the ciphertexts and keys.

Such comparisons must be weighed according to the intended application. For example, computation time is often a greater concern than ciphertext size, except for wireless and bandwidth-constrained applications. Also, certain applications such as email are by nature one-to-many, in which case any scheme that supports multiple recipients will have an advantage over its competitors.

5.4.1 Space Efficiency

Table 5 compares the space overheads of the various data types for each scheme. The most significant overhead to consider is the ciphertext's, followed by the public system parameters and/or the users' private keys, depending on the application. The master key sizes are also listed for completeness.

Ciphertext overheads are reported discounted of the size of the message itself, *i.e.*, we subtract from the ciphertext size the number of message bits that are freely chosen by the sender with the intent of being recovered by the recipient (for KEM, the message size is considered to be zero, since the session key is random and not effectively chosen by the sender). This makes it possible to make direct comparisons between IBE and IBKEM, since an IBE can be used by itself whereas an IBKEM is useless without a DEM (whose overhead will have to be added to the total).

In practice one may have to make different corrections. As we noted before, IBEs are often used indirectly, not to encrypt messages, but merely to encrypt session keys, in which case the entire IBE ciphertext will constitute overhead. For IBKEMs, the overhead will further increase in multi-recipient applications, where an interface layer between KEM and DEM must be used so that the same session key can be used for all the recipients (to allow sharing of the DEM ciphertext).

5.4.2 Time Efficiency

Table 6 shows a tally of the number of group operations required of each scheme. Separate counts are given for each of the three groups $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$, and a distinction is made between general exponentiations (*i.e.*, raising an arbitrary base to an arbitrary exponent) and exponentiations with a fixed base (to an arbitrary exponent). The distinction is important since fixed-base exponentiations may be optimized to incur a much smaller amortized cost. As a general rule, the factors mostly affecting computational costs are:

1. the number of *independent* pairings (with all known algorithms, a product or ratio of two pairings is only slightly more expensive than a single pairing); and, to a lesser extent:

2. the number of *general* exponentiations, *i.e.*, subexpressions of the form b^x where the base b is not known ahead of time (by contrast, fixed-base exponentiations can be calculated much faster after investing in a moderate amount of pre-computation);

3. the number of hashing operations into either bilinear group $\mathbb{G}$ or $\hat{\mathbb{G}}$ (hashing to a string or into $\mathbb{Z}_p$ is virtually free by comparison).

6 CURVES AND PERFORMANCE

To illustrate how the schemes' time and space complexities will compare in practice, it is useful to focus on a few concrete elliptic curve implementations.

Table 4: Extended functionalities known to be compatible with the BF, SK, and BB_1 frameworks.

	Full IBE			IBKEM Only		
	BF-IBE	SK-IBE	BB_1-IBE	BF-KEM	SK-KEM	BB_1-KEM
Multi-Recipient	✓	✓	✓	$\sim^*$	$\sim^*$	$\sim^*$
Streaming			$✓^{**}$	$✓^{**}$	$✓^{**}$	$✓^{**}$
Master-Key Sharing						
n-out-of-n	✓	$\sim^{***}$	✓	✓	$\sim^{***}$	✓
t-out-of-n	✓	$\sim^{***}$	✓	✓	$\sim^{***}$	✓
Hierarchical Identities						
linear-size CT	✓	✓	✓	✓	✓	✓
const.-size CT			✓			✓
Forward Security						
list (lin-size)	✓	✓	✓	✓	✓	✓
tree (log-size)	✓		✓	✓		✓
compr'd ($<$ log)			✓			✓

* Multi-recipient KEM requires an additional intermediate session key layer if the DEM component is reused.

** Secure streaming requires that the recipient application can be instructed to reject a stream after decryption.

*** Secret sharing is possible in SK using new techiques (Boyen, 2007) at the expense of longer $\Theta(n)$-size ciphertexts.

Table 5: Size requirements for the various data types in the stated instances of BF, SK, and BB_1. The size of each data type is expressed by the number of elements in $\mathbb{Z}_p$, $\mathbb{G}$, $\hat{\mathbb{G}}$, $\mathbb{G}_t$, and ℓ-bit hash output strings needed for its representation.

	Full IBE			IBKEM Only		
	BF-IBE	SK-IBE	BB_1-IBE	BF-KEM	SK-KEM	BB_1-KEM
System Parameters						
# elts. $\in \mathbb{G}$	2	2	3	2	2	3
# elts. $\in \mathbb{G}_t$	0	1	1	0	1	1
Master Secret						
# elts. $\in \mathbb{Z}_p$	1	1	3	1	1	3
# elts. $\in \hat{\mathbb{G}}$	0	0	1	0	0	1
Private Key						
# elts. $\in \hat{\mathbb{G}}$	1	1	2	1	1	2
Ciphertext*						
# elts. $\in \mathbb{Z}_p$	0	0	1			
# elts. $\in \mathbb{G}$	1	1	2			
# hash strs.	1	1	0			
KEM Capsule						
# elts. $\in \mathbb{Z}_p$				0	0	0
# elts. $\in \mathbb{G}$				2	1	1
# hash strs.				0	1	0

* The ciphertext overhead for encryption excludes the size of the sender-selected message.

Table 6: Time requirements for the various operations in the listed instances of BF, SK, and BB_1. Each cryptographic algorithm is decomposed into an optimal sequence of algebraic calculations. Each calculation is then denoted by its type and the algebraic group in which it is performed.

	Full IBE			IBKEM Only		
	BF-IBE	SK-IBE	BB_1-IBE	BF-KEM	SK-KEM	BB_1-KEM
Private Key Extraction						
# fix-base exp.		$\hat{\mathbb{G}}$	$\hat{\mathbb{G}}\ \hat{\mathbb{G}}$		$\hat{\mathbb{G}}$	$\hat{\mathbb{G}}\ \hat{\mathbb{G}}$
# general exp.	$\hat{\mathbb{G}}$			$\hat{\mathbb{G}}$		
# hashes	$\hat{\mathbb{G}}$			$\hat{\mathbb{G}}$		
Encryption						
# fix-base exp.	$\mathbb{G}\ \mathbb{G}$	$\mathbb{G}\ \mathbb{G}\ \mathbb{G}_t$	$\mathbb{G}\ \mathbb{G}\ \mathbb{G}\ \mathbb{G}_t$			
# hashes	$\hat{\mathbb{G}}$					
# pairings	$\to \mathbb{G}_t$					
Decryption						
# fix-base exp.	$\mathbb{G}$	$\mathbb{G}\ \mathbb{G}$	$\mathbb{G}\ \mathbb{G}_t$			
# pairings	$\to \mathbb{G}_t$	$\to \mathbb{G}_t$				
# pairing ratios			$\to \mathbb{G}_t$			
Encapsulation						
# fix-base exp.				$\mathbb{G}\ \mathbb{G}$	$\mathbb{G}\ \mathbb{G}\ \mathbb{G}_t$	$\mathbb{G}\ \mathbb{G}\ \mathbb{G}\ \mathbb{G}_t$
# hashes				$\hat{\mathbb{G}}$		
# pairings				$\to \mathbb{G}_t$		
Decapsulation						
# fix-base exp.					$\mathbb{G}\ \mathbb{G}\ \mathbb{G}_t$	
# pairings				$\to \mathbb{G}_t$	$\to \mathbb{G}_t$	
# pairing ratios						$\to \mathbb{G}_t$

6.1 Elliptic Curve Selection

We consider three combinations of curve types and security parameters: supersingular (SS) curves of embedding degree 2 over large prime fields at security levels 80 and 128, and non-supersingular curves of embedding 6 at security level 128 called MNT curves.

SS curves provide the most natural instances of type-1 pairings, and are the ones originally used in the Boneh-Franklin IBE system. We only consider SS curves over large-characteristic fields. These have the inconvenient of pegging the embedding degree to the very small value 2, which is inefficient at large security levels; however, they have the advantage of being plentiful, easy to sample, and quite simple to implement. Additional technical details may be found in Boneh and Franklin (2001).

MNT curves can be used to construct natural type-2 or type-3 bilinear groups: BF can be implemented on the latter, SK on the former, and BB_1 indifferently on either type. MNT curves come in several flavors; here we consider the constructions with the largest embedding degree, equal to 6, which are probably the most useful. These curves were first proposed by Miyaji et al. (2001), and have been used in the signature scheme of Boneh et al. (2004b).

Two competing factors intervene when choosing appropriate parameters for the curves.

1. At security level σ, the curve order must have a large prime factor $p > 2^{2\sigma-1}$, which will enable us to construct cyclic groups $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$ of prime order p large enough to defeat generic discrete-logarithm attacks. *E.g.*, for $\sigma = 80$, we need $p \approx 2^{160}$ or bigger.

2. Second, the curve must be constructed over a finite field (which we assume has prime order) that is large enough to defeat the best known discrete-logarithm attacks in the appropriate field extension. *I.e.*, for a prime field $\mathbb{F}_q$, the extension $\mathbb{F}_{q^k}$ must be large enough to defeat the Number Field Sieve, where k is the embedding degree of the p-order subgroup of the selected curve E in $\mathbb{F}_q$. Concretely, at security levels $\sigma = 80$ and $\sigma = 128$, we respectively need $q^k \approx 2^{1024}$ and $q^k \approx 2^{3072}$, which for $k = 6$ gives us $q \approx 2^{171}$ and $q \approx 2^{512}$ respectively.

Choices of curves that let us satisfy both constraints tightly at the same time will typically result in more economical implementations with more compact representations, for a given security level.

Table 7 shows the representation sizes for group elements for the three combinations of curves and security parameters. These numbers are derived from the intrinsic security requirements of the curves, and do not take into account security losses that might arise in actual cryptographic schemes.

Table 8 gives rough estimations of the relative costs of the various algebraic operations in the three groups $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$, for each combination of curve type and security parameter. The numbers are all set on the same (arbitrary) scale, and can thus be compared with each other as a first approximation. These numbers are however merely indicative, since in practice, actual running times will depend on many factors, *e.g.*, internal representations, the choice of pairing, exponentiation algorithms, space/time trade-offs, CPU types, memory constraints, caching, and a number of other factors.

6.2 Complexity and Overhead

For illustrative value, we now quantify the space and time requirements of the various schemes for the various curves and security parameters we selected.

Table 9 lists the representation overheads for the BF, SK, and BB_1 frameworks using the three choices of curves and security levels given above; the sizes are expressed in bits.

Table 10 lists the estimated relative running times for the various systems in the same conditions; the indicative computational costs are expressed in arbitrary units.

Caveat. Tables 9 and 10 are based on the intrinsic security of the various curves, and are oblivious to all security losses caused by the reductionist proofs of the different schemes. These losses vary with circumstances and will require non-negligible corrections on a case-by-case basis.

6.3 Correcting for Exact Security

Since none of the known IBE schemes has provable security parameters that are independent of the number of adversarial queries, the various schemes may be less secure in practice than the idealized Tables 8 and 10 suggest. For any given curve and group size, the "exact security" of each scheme will be negatively affected by more-or-less severe polynomial slack factors that were hidden in the asymptotic statements of Theorems 1, 2, and 3. (The unabridged theorem statements can be found in the referenced works.) A precise analysis of the security losses induced by these slack factors falls beyond the scope of this note; nevertheless, a number of general observations can be made:

- BB_1-IBE has a security reduction to BDH which degrades in $O(1/q_{H_1})$; *i.e.*, the loss factor is linear in the number of random-oracle queries made to H_1, and virtually independent of the number of decryption, key extraction, and other random-oracle queries.

- BF-IBE has a security reduction that degrades in $O(1/q_{H_2}(q_{H_3} + q_{H_4})q_K)$; *i.e.*, security degrades quadratically in the total number of random-oracle queries and linearly in the number of key extraction queries.

- BB_1-IBKEM and BF-IBKEM have respective security reductions in $O(1/q_{H_1})$ and $O(1/q_K)$, but because these KEMs do not provide redundancy for explicit ciphertext rejection, the reduction is to the Gap version of BDH, which is a much stronger assumption than computational BDH.

- SK-IBE and SK-IBKEM have by far the worst security reductions of all the schemes that we described. Specifically, the reduction efficiency of both SK schemes are a function of $O(1/q_{H_1}q_{H_2}(q_{H_3} + q_{H_4}))$, which is to say that security degrades with the *cube* of the total number of random-oracle queries. The KEM uses the same assumption as the encryption scheme, because both use essentially the same explicit ciphertext authentication method.

- In addition to the losses caused by their security reductions, SK-IBE and SK-IBKEM also suffer from security losses that are intrinsic to the q-BDHI assumption. These losses stem from the fact that q-BDHI problem instances are not of constant size but of size $\propto q \geq q_K$, which can be quite large in practice if the system cannot be otherwise defended against active attacks. Such "large-q" assumptions are used in all known exponent-inversion schemes, which, besides SK, also include BB_2 and Gentry's IBE.

  Specifically, the lower complexity bounds of Boneh and Boyen (2004b) and the corresponding upper bounds of Cheon (2006) imply that, in the generic-group model, the concrete security of q-BDHI instances against discrete-log attacks ranges between $1/q$ and $1/q^3$ that of the corresponding BDH instance (the actual security degradation depends on the number of generic-group oracle queries that the adversary will otherwise be making). It follows that the generic security of SK-IBE and SK-IBKEM in the sense of Shoup (1997) has rather rapidly decreasing bounds between $O(1/q_H^3 q_K)$ on the up-side and $\Omega(1/q_H^3 q_K^3)$ on the down-side. As before, q_H and q_K are the number of random-oracle and key-extraction queries made by the adversary.

To compensate for such concrete security losses, the general rule is to boost the "apparent" security parameter by a suitable coefficient to raise the "actual" concrete security of the scheme up to the prescribed level. The good news is that polynomial security losses are relatively inexpensive to compensate in this manner, because security

Table 7: Representation sizes, in bits, for the various group elements on different types of elliptic curves at usual security levels. These are practical representation sizes with simple optimizations such as "point compression" for the representation of elliptic curve elements (for $\mathbb{G}$ and $\hat{\mathbb{G}}$).

	Representation Sizes (bits)		
	SS @ 80-bit security	MNT @ 80-bit security	MNT @ 128-bit security
$\mathbb{Z}_p$	160	160	256
$\mathbb{G}$	512	171	512
$\hat{\mathbb{G}}$	512	1026	3072
$\mathbb{G}_t$	1024	1026	3072

Table 8: Estimated calculation times for various algebraic operations on the same elliptic curves at the same security levels as in Table 7. The time unit is defined as the cost of a general exponentiation (*i.e.*, point multiplication) on a random 171-bit elliptic curve for a random 160-bit exponent. Timings are indicative only, and do not account for any exact-security reduction inefficiencies that depend on the number of adversarial queries.

	Relative Timings (arbitrary unit)*		
	SS @ 80-bit security	MNT @ 80-bit security	MNT @ 128-bit security
In $\mathbb{G}$:			
fix-base expon.	2	0.2	3
general expon.	10	1	15
In $\hat{\mathbb{G}}$:			
fix-base expon.	2	8	100
general expon.	10	40	500
hashing	10	40	500
In/to $\mathbb{G}_t$:			
fix-base expon.	2	2	30
general expon.	10	10	150
single pairing	100	100	1500
ratio of pairings	120	120	1800

* Unit = point multiplication time on random curve $E/\mathbb{F}_q$ by random scalar in $\mathbb{Z}_p$, for prime $q \approx 2^{171}$ and $p \approx 2^{160}$.

increases super-polynomially with the number of "cryptographic bits". Unfortunately, there is no universal correction recipe, because the security losses to recover often depends on application-specific operating conditions, such as how much access to a decryption oracle an adversary can be expected to gain in the most pessimistic scenario. The suitable correction coefficient must therefore be determined on a case-by-case basis.

6.4 General Recommendations

We saw that it is often necessary to raise the *apparent security parameters* in order to compensate for security reduction inefficiencies that are function of the adversary's behavior. How much adjustment is needed will thus depend on the powers of the worst-case adversaries that one is willing to defend against. This varies from one context to the next, but we can make the following general recommendations based on our observations:

The SK schemes should generally be avoided as a rule of thumb, except perhaps in very special circumstances where simplicity of implementation is the absolute overriding concern.

The BF schemes are safe to use, but are penalized by the lack of efficiency of their encryption and encapsulation algorithms. BF encryption is safer than BF key encapsulation the way we described it, because the latter requires a stronger Gap assumption; this was the price to pay for its extreme compactness.

The BB_1-IBE scheme appears to be the smartest choice, due to a combination of operational advantages, a fairly efficient security reduction, and its reliance on a reasonable complexity assumption. In addition, the commutative-blinding framework of BB_1 has the advantage of offering some *residual security* outside of the random-oracle model, without any change to the

Table 9: Actual overheads, in bits, for the various IBE and IBKEM schemes, in function of the type of elliptic curve and the security level.

Total Overhead (bits)	Full IBE			IBKEM Only		
	BF-IBE	SK-IBE	BB_1-IBE	BF-KEM	SK-KEM	BB_1-KEM
SS @ 80-bit security level						
public params.	1024	2048	2560	1024	2048	2560
CT (excl. msg.)	672	672	1184			
KEM capsule				512	672	1024
MNT @ 80-bit security level						
public params.	342	1368	1539	342	1368	1539
CT (excl. msg.)	331	331	502			
KEM capsule				171	331	342
MNT @ 128-bit security level						
public params.	1024	4096	4608	1024	4096	4608
CT (excl. msg.)	768	768	1280			
KEM capsule				512	768	1024

Table 10: Estimated relative computational costs, in arbitrary units, for the various IBE and IBKEM schemes, in function of the type of elliptic curve and the security level. Timings are indicative only, and the same caveats as in Table 8 apply.

Computation Cost*	Full IBE			IBKEM Only		
	BF-IBE	SK-IBE	BB_1-IBE	BF-KEM	SK-KEM	BB_1-KEM
SS @ 80-bit security level						
key extraction	20	2	4	20	2	4
encryption	114	6	8			
decryption	102	104	124			
encapsulation				114	6	8
decapsulation				100	106	120
MNT @ 80-bit security level						
key extraction	80	8	16	80	8	16
encryption	140.4	2.4	2.6			
decryption	100.2	100.4	122.2			
encapsulation				140.4	2.4	2.6
decapsulation				100	102.4	120
MNT @ 128-bit security level						
key extraction	1000	100	200	1000	100	200
encryption	2006	36	39			
decryption	1503	1506	1833			
encapsulation				2006	36	39
decapsulation				1500	1536	1800

* Estimated indicative times in the same time unit as in Table 8, and for comparison purposes only.

system, as discussed by Boneh and Boyen (2004a, §7).

Because of its more efficient reduction, BB_1-IBE is also likely to require a smaller group size $p = |\mathbb{G}| = |\hat{\mathbb{G}}| = |\mathbb{G}_t|$ than BF and especially SK, in order to meet prescribed exact security levels against real-world adversaries. For this reason, and in spite of its apparent complexity, BB_1-IBE will thus quite possibly have the shortest ciphertext and the fastest operation, depending on the application.

ACKNOWLEDGEMENT

The author wishes to thank Eike Kiltz and Shengbao Wang for comments on an earlier draft related to this note.

REFERENCES

Bentahar, K., Farshim, P., Malone-Lee, J., and Smart, N. P. (2005). Generic constructions of identity-based and certificateless kems. Cryptology ePrint Archive, Report 2005/058. `http://eprint.iacr.org/`.

Blake, I., Seroussi, G., and Smart, N. (1999). *Elliptic Curves in Cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press.

Blake, I. F., Seroussi, G., and Smart, N. P., editors (2005). *Advances in Elliptic Curve Cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*. Cambridge University Press.

Boneh, D. and Boyen, X. (2004a). Efficient selective-ID secure identity based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–38. Springer-Verlag.

Boneh, D. and Boyen, X. (2004b). Short signatures without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer-Verlag.

Boneh, D., Boyen, X., and Goh, E.-J. (2005). Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–56. Springer-Verlag.

Boneh, D., Boyen, X., and Halevi, S. (2006). Chosen ciphertext secure public key threshold encryption without random oracles. In *Proceedings of RSA-CT 2006*, volume 3860 of *LNCS*, pages 226–43. Springer-Verlag.

Boneh, D., Boyen, X., and Shacham, H. (2004a). Short group signatures. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer-Verlag.

Boneh, D. and Franklin, M. (2001). Identity-based encryption from the Weil pairing. In *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–29. Springer-Verlag.

Boneh, D., Gentry, C., and Hamburg, M. (2007). Space-efficient identity based encryption without pairings. Cryptology ePrint Archive, Report 2007/177. `http://eprint.iacr.org/`.

Boneh, D., Lynn, B., and Shacham, H. (2004b). Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319.

Boyen, X. (2007). General ad hoc encryption from exponent inversion IBE. In *Advances in Cryptology—EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 394–411. Springer-Verlag.

Canetti, R., Halevi, S., and Katz, J. (2003). A forward-secure public-key encryption scheme. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *LNCS*. Springer-Verlag.

Canetti, R., Halevi, S., and Katz, J. (2004). Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*. Springer-Verlag.

Chatterjee, S. and Sarkar, P. (2005). Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In *Proceedings of ICISC 2005*.

Chen, L. and Cheng, Z. (2005). Security proof of Sakai-Kasahara's identity-based encryption scheme. Cryptology ePrint Archive, Report 2005/226. `http://eprint.iacr.org/`.

Chen, L., Cheng, Z., Malone-Lee, J., and Smart, N. P. (2005). An efficient ID-KEM based on the Sakai-Kasahara key construction. Cryptology ePrint Archive, Report 2005/224. `http://eprint.iacr.org/2005/224/`.

Cheon, J. H. (2006). Security analysis of the Strong Diffie-Hellman problem. In *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11. Springer-Verlag.

Cocks, C. (2001). An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*.

Cramer, R. and Shoup, V. (1998). A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology—CRYPTO 1998*, volume 1462 of *LNCS*. Springer-Verlag.

Cramer, R. and Shoup, V. (2002). Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology—EUROCRYPT 2002*, volume 2729 of *LNCS*, pages 45–64. Springer-Verlag.

Galbraith, S., Paterson, K., and Smart, N. (2006). Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165. `http://eprint.iacr.org/`.

Gentry, C. (2006). Practical identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2006*, LNCS. Springer-Verlag.

Gentry, C. and Silverberg, A. (2002). Hierarchical ID-based cryptography. In *Proceedings of ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 548–66. Springer-Verlag.

Halevi, S. and Rogaway, P. (2003). A tweakable enciphering mode. In *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–99. Springer-Verlag.

Halevi, S. and Rogaway, P. (2004). A parallelizable enciphering mode. In *Topics in Cryptology—CT-RSA 2004*, LNCS, pages 292–304. Springer-Verlag.

Horwitz, J. and Lynn, B. (2002). Towards hierarchical identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2002*, LNCS, pages 466–81. Springer-Verlag.

Joux, A. (2004). A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–76.

Joux, A. and Nguyen, K. (2003). Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–47.

Libert, B. and Quisquater, J.-J. (2005). Identity based encryption without redundancy. In *Proceedings of ACNS 2005*, volume 3531 of *LNCS*, pages 285–300. Springer-Verlag.

Lynn, B. (2007). *On the Implementation of Pairing-Based Cryptosystems*. PhD thesis, Stanford University.

Maurer, U. M. and Yacobi, Y. (1996). A non-interactive public-key distribution system. *Designs, Codes and Cryptography*, 9(3):305–16.

Miller, V. (2004). The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4).

Mitsunari, S., Sakai, R., and Kasahara, M. (2002). A new traitor tracing. *IEICE Transactions on Fundamentals*, E85-A(2):481–4.

Miyaji, A., Nakabayashi, M., and Takano, S. (2001). New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals*, E84-A(5):1234–43.

Naccache, D. (2005). Secure and practical identity-based encryption. Cryptology ePrint Archive, Report 2005/369. `http://eprint.iacr.org/`.

Naor, M. (2003). On cryptographic assumptions and challenges. In *Advances in Cryptology—CRYPTO 2003*, LNCS, pages 96–109. Springer-Verlag.

Naor, M. and Yung, M. (1990). Public-key cryptosystems provably secure against chosen ciphertext attacks. In *ACM Symposium on Theory of Computing—STOC 1990*, pages 427–37. ACM Press.

Sakai, R. and Kasahara, M. (2003). ID based cryptosystems with pairing over elliptic curve. Cryptology ePrint Archive, Report 2003/054. `http://eprint.iacr.org/2003/054/`.

Sakai, R., Ohgishi, K., and Kasahara, M. (2000). Cryptosystems based on pairings. In *Symposium on Cryptography and Information Security—SCIS 2000*, Japan.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Advances in Cryptology—CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag.

Shoup, V. (1997). Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology—EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 256–66. Springer-Verlag.

Shoup, V. (2000). Using hash functions as a hedge against chosen ciphertext attack. In *Advances in Cryptology—EUROCRYPT 2000*, LNCS. Springer-Verlag.

Shoup, V. (2001). A proposal for an ISO standard for public key encryption (version 2.1). Manuscript. Available from `http://shoup.net/papers/`.

Steiner, M., Tsudik, G., and Waidner, M. (1996). Diffie-Hellman key distribution extended to groups. In *ACM Conference on Computer and Communications Security—CCS 1996*.

Tanaka, H. (1987). A realization scheme for the identity-based cryptosystem. In *Advances in Cryptology—CRYPTO 1987*, volume 293 of *LNCS*, pages 341–49. Springer-Verlag.

Tsujii, S. and Itoh, T. (1989). An ID-based cryptosystem based on the discrete logarithm problem. *IEEE Journal on Selected Areas in Communication*, 7(4):467–73.

Waters, B. (2005). Efficient identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*. Springer-Verlag.

Yao, D., Fazio, N., Dodis, Y., and Lysyanskaya, A. (2004). ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In *ACM Conference on Computer and Communications Security—CCS 2004*, pages 354–63. ACM Press.