The BB_1 Identity-Based Cryptosystem :
A Standard for Encryption and Key Encapsulation

Xavier Boyen
Voltage Inc.

August 14, 2006
IEEE P1363.3 draft

1 Standardization Rationale

This note describes and discusses a couple of concrete instantiations of the "BB_1" identity-based cryptosystem proposed at Eurocrypt 2004 by Boneh and Boyen. The emphasis is on practicality, performance, flexibility, and standardization.

1.1 Purpose and Scope

The main purpose of this document is to provide simple and high-level, yet actionably, concrete, and standard descriptions of two practical instantiations of the Boneh-Boyen BB_1 identity-based cryptosystem. One instantiation is an full encryption (or IBE) system, the other a key encapsulation mechanism (or IBKEM).

A secondary purpose of the document is to discuss the many extensions of practical interest that the BB_1 scheme allows, and provide detailed comparisons with other schemes published in the literature.

Selected Scheme. To dispell any ambiguity, we recall that Boneh and Boyen in their original paper [BB04] offered two distinct identity-based encryption schemes, based on very different sets of methods and assumptions. The present document focuses exclusively on (instantiations of) Boneh and Boyen's *first* IBE scheme — *cfr.* §4, *op. cit.* [BB04] —, which we (collectively) denote "BB_1".

Practical Focus. For reasons of practicality and simplicity, we mainly limit our attention to flat (non-hierarchical) instantiations of BB_1, and with hashed identities in the random oracle model. This is in contrast with the academic article, which sought maximal generality and provable security at the expense of practical optimization.

On the other hand, in order to facilitate the use of this note as an implementation guide, we give explicit instantiations of both Identity-Based Encryption (IBE) and Identity-Based Key Encapsulation Mechanism (IBKEM), along with the necessary security measures to withstand adaptive chosen-identity and adaptive chosen-ciphertext attacks (in the random oracle model).

Mathematics. We will make minimal mention of notions of elliptic curves and pairings. These concepts are needed in order to implement BB_1 schemes in practice; on the other hand, the clear separation between the cryptographic constructs and the mathematical objects upon which they are built make it relatively easy to treat the latter as a black box, and leave the details to the choice of the implementer.

Formal Proofs. We often omit security proofs and only briefly mention underlying complexity assumptions. Formal security reductions are extremely important when comparing the security of competing proposals, but remain of limited interest in a practical guide such as this. We merely note that the random oracle instantiations of BB_1 described here have equal or better "exact security" than every other identity-based encryption known at the date of this writing, for the security notions alluded to above (and several others), for any fixed security level and choice of elliptic curve implementation.

1.2 Organization of the Document

In the following two sections, we give a rationale for standardization on the proposed instantiations of the BB_1 cryptosystem. In Section 2, we start by offering a classification of all IBE systems known to date, to help us understand how these systems differ. In Section 3, we argue that the BB_1 cryptosystem enjoys a large number of benefits over all the other identity-based cryptosystems that have been proposed to date.

The technical presentation of the proposal will follow in Sections 4 and 5. For reference purposes, factual comparisons with other schemes will be made in Appendix A.

2 Classification of IBE Schemes

The following is a rough classification of the known identity-based encryption schemes. All of them support at least a basic security reduction to a well-formulated complexity assumption, either in the standard model or in the random oracle model.

We remark that although arguments in idealized models such as the random oracle model carry less weight than proofs in the standard model, both are equally adequate to provide confidence in a scheme, in the context of a standardization effort geared toward flexibility and practicality.

2.1 "Quadratic Residuosity" IBE (without pairings)

The only known IBE system that does not use bilinear pairings is due to Cocks [Coc01]. The Cocks system relies on the hardness of the quadratic residuosity problem — which is that of determining whether $\exists y : x = y^2 \pmod{N}$, given a modular residue $x \in \mathbb{Z}_N$ and a composite modulus $N = p_1 p_2$.

The Cocks system is reasonably fast, but is very bandwidth consuming as it requires $\log N$ bits of ciphertext per bit of plaintext. Additionally, it is not known to be provably secure against adaptive identity attacks.

2.2 "Full Domain Hash" IBE

The first practical IBE system was proposed by Boneh and Franklin [BF01]. It uses pairings, and relies on a very reasonable complexity assumption. The most important factor that contributed to its popularity is its efficiency and its much reduced bandwidth requirement, compared to the Cocks system that came out at about the same time.

The drawback of the BF approach is that it makes extensive use of cryptographic hashes (modeled as random oracles), and in particular assumes the availability of uniformly distributed hash functions with images in the pairing group (*i.e.*, on an elliptic curve), as in: $h = H(\mathsf{ID}) \in \mathbb{G}$, where ID is a string, and $\mathbb{G}$ the bilinear group. Two problems with this kind of hashing is that it is somewhat expensive, and more importantly that it restricts our choice of curves (or forces us to use the curves we have less efficiently). These factors contribute to an overall reduction of efficiency in the final deployment.

2.3 "Exponent Inversion" IBE

Following an idea of Mitsunary, Sakai, and Kasahara in the context of tracing [MSK02], a number of IBE systems have been proposed that rely on the general idea that exponents are difficult to invert (*i.e.*, it is hard to compute $g^{1/x}$ given g and g^x), and thus that with a pairing one can selectively "cancel out" exponents without revealing them (by giving out $g^{1/x}$). In the context of IBE, the goal is to encode identities as part of this x, and thus avoid the need to hash directly on the curve.

Unfortunately, proving the security of such schemes is not a simple affair. The first scheme based on this approach, due to Sakai and Kasahara [SK03], was proposed without security proof. The first provably secure IBE scheme to use the inversion idea was given by Boneh and Boyen [BB04] (being the BB_2 system in the same paper as BB_1 but not to be confused with it). The BB_2 system was designed for the purpose of achieving security without random oracles. Later, a proof for a variant of the original SK system was finally given, by Chen *et al.* [CCMLS05], essentially by introducing a random oracle in the earlier BB_2 proof. Recently, Gentry [Gen06] proposed a clever variation on the general theme, with a tighter security proof than the earlier proposals.

A common characteristic of all systems based on the "exponent inversion" principle is that their security proofs require surprisingly strong assumptions. A typical assumption states that it is hard to compute $g^{1/x}$ (or $e(g,g)^{1/x}$) given not only g and g^x, but also g^{x^2}, g^{x^3}, all the way up to g^{x^q}. (Gentry's scheme uses an even stronger assumption.) In an IBE scheme, q would be the maximum number of private key owners that an adversary may corrupt in an active attack. The value of q must be a fairly large number for the proofs to have any meaning — but then the assumption becomes correspondingly less trustworthy.

In fact, the above and other related assumptions have recently come under (limited) number-theoretic attack in a recent paper by Cheon [Che06]. This should serve as a warning, and remind us to use a healthy dose of caution if one were to deploy any of the IBE systems in this category.

2.4 "Commutative Blinding" IBE

The last category of IBE systems was initiated by Boneh and Boyen [BB04] with their BB_1 scheme. Their approach sidesteps most or all the problems associated with the BF and the SK/BB_2 approaches: in particular, the Boneh-Boyen security reduction uses the same weak assumption as Boneh-Franklin, while allowing identities to be encoded as integers rather than hashed on the curve. Very roughly, the BB_1 idea is to create blinding factors with two or more secret coefficients, that "commute" (*i.e.*, that can be applied in either order), thanks to the pairing.

Perhaps the best quality of this paradigm is exemplified by the great flexibility of the BB_1 algebraic structure. The basic system already supported a number of extensions found in BF, but that are not known to be possible with SK/BB_2/Gentry. These extensions include Master-Key Sharing ("Threshold"), Hierarchical Identities ("HIBE"), and Forward Security, all of which are directly relevant to the practitioner.

Many small and large modifications to the basic scheme have been proposed over the past two years, sometimes with surprising properties. For example, Sahai and Waters' [SW05] "Fuzzy" IBE was created by replacing the recipient identity by an error correcting code in the BB_1 ciphertext. In a related result, Waters [Wat05] showed that a minor modification to BB_1 enabled it to support a much stronger security notion outside of the random oracle model. Subsequent improvements of Waters' proof have also been proposed in [CS05] and [Nac05].

At the other end of the spectrum, Boyen and Waters [BW06] recently created an Anonymous (H)IBE scheme that relies on new techniques and a different assumption. Despite its limited resemblance to the original BB_1 scheme, its crucial use of commutative blinding exponents also earns it a place in this category.

3 Competitive Features of BB_1

We now briefly turn to the technical reasons that motivated our choice of BB_1 as the best known platform for IBE and IBKEM, and in particular the random oracle instantiations of BB_1 described later in this document (following which we shall revisit this issue and give a more precise, point-by-point comparison with the competing schemes).

3.1 Hardness Assumptions

The BF and BB_1 schemes have in common that they rely on the Bilinear Diffie-Hellman (BDH) complexity assumption. The BDH assumption is non-interactive, and is naturally and concisely stated as follows: given g, g^a, g^b, g^c, it is hard to compute $e(g,g)^{abc}$. It is arguably one of the mildest assumptions in the rapidly populating zoo of pairing-related assumptions.

By contrast, the SK/BB_2/Gentry schemes rely on a class of "exponent inversion" parameterized by some large integer q. For example, the q-BDHI assumption posits the hardness of computing $e(g,g)^{1/x}$ given g and g^{x^i} for all $i = 1, \ldots, q$. This is problematic in several respects. On the one hand, the parameter q and the very long problem instances are barriers to the publication of standard challenges (such as the RSA challenges), which is likely to discourage falsification efforts and therefore procure a false sense of security. On the other hand, the large amount of data creates theoretical vulnerabilities of its own. For example, and notwithstanding the above, a previously unknown number-theoretic attack against the whole class has just been published in the past few months [Che06]. Although this particular vulnerability does not pose a catastrophic threat to any of the SK/BB_2/Gentry schemes, it requires an adjustment to the parameters that is detrimental to efficiency.

3.2 Flexibility

There are two aspects of flexibility to be considered: "mathematical compatibility", which captures the feasibility of building a primitive from abstract mathematical objects; and "application versatility", which indicates what can ultimately be done with a particular primitive. We analyze both in turn.

3.2.1 Mathematical Compatibility

Bilinear pairings come in three flavors depending on the curves on which they are defined [GPS06]. Type-1 pairings are symmetric functions of the form $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$. Type-2 and type-3 pairings are asymmetric functions $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$; the difference between them is that in the type-2 case there is a known homomorphism $\phi : \hat{\mathbb{G}} \to \mathbb{G}$; while in the type-3 case there is no efficient way to move between the groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ in either direction. In addition to the homomorphism, there is the issue of hashing. Some curves support easy sampling and/or hashing (the usual case for type-1), while others may not, or only in one of the groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ but not in the other. For example, known type-2 curves preclude hashing into $\hat{\mathbb{G}}$, though type-1 and type-3 curves allow hashing into both groups. For maximum compatibility, it is therefore desirable to avoid hashing in any of $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$. (Hashing into a known interval of $\mathbb{Z}$ is a much simpler operation.)

All pairing-based primitives and protocols are not created equal. Some require type-1 pairings for their ability to swap arguments and make Diffie-Hellman tuples easy to decide. Others expressly forbid that; such is the case of constructions that assume the DDH problem to be hard in $\mathbb{G}$, or even simultaneously in $\mathbb{G}$ and $\hat{\mathbb{G}}$. Orthogonally to that, one should consider whether a given construction requires sampling or hashing into $\mathbb{G}$, $\hat{\mathbb{G}}$, or possibly $\mathbb{G}_t$. The least demanding of all schemes are indifferent to the presence of an efficient homomorphism, do not require hashing, and furthermore can take advantage of a disparity in representation size between elements of $\mathbb{G}$ and of $\hat{\mathbb{G}}$ to minimize the bandwidth requirements.

BB_1 fulfills all the above criteria of maximum compatibility:

- no homomorphism required (in either direction, either in the scheme or the security reduction);
- no hashing required (other than into intervals of $\mathbb{Z}$).

BB_2 (and SK with the BB_2 security reduction) is not directly compatible with type-3 curves because the security reduction requires a homomorphism $\phi : \hat{\mathbb{G}} \to \mathbb{G}$, unless one is willing to make even stronger assumptions.

BF is the most demanding scheme of the lot, as it requires homomorphism and hashing.

3.2.2 Application Versatility

The other aspect of a cryptographic construction's flexibility concerns the ease with which it can be extended and adapted to suit one's particular application. For an IBE standard, certain capabilities are almost a requirement:

- threshold secret sharing (for the master key and/or the private keys);
- hierarchical identities (for organizational hierarchies or to support delegation of authority);
- forward security (sufficiently efficient to support finely granular time periods).

The ability to split the master secret into shares held in distinct locations may be crucial to alleviate concerns about the "unavoidable" key escrow implied by the definition of IBE. Hierarchical identities are also desirable in certain applications, such as for large compartmented organizations. Forward security is desirable to ensure that a catastrophic security breach such as exposure of the master secret does not compromise the confidentiality of earlier encryptions. The above capabilities are all easy to achieve in the BB_1 framework, and to a lesser extent in the BF system. However, it is not known how to do either of them in the SK/BB_2/Gentry systems.

Another factor to weigh when considering versatility is the promise of future extensions. We have already hinted at the remarkable versatility of the BB_1 framework, due to its algebraic structure. Useful extensions of BB_1 in the broad sense that have recently been proposed include:

- aggressive ciphertext compression (in hierarchies as well as in forward secure systems);
- anonymity (regarding the recipient identity), with applications to search on encrypted data.

Indications abound that "Commutative Blinding" is a more versatile IBE framework than "Full Domain Hash", per our earlier classification. Indeed, features such as those listed above have been achieved more elegantly in BB_1 than in BF, where they typically come with additional restrictions. As for the "Exponent Inversion" framework, without much doubt it is the least flexible of all. This is evidenced by the fact that no counterparts to the above features seem to exist in SK, BB_2, or Gentry's IBE; conversely, there is no functionality available to these systems that is not easy to replicate in BB_1.

3.3 Time and Space Efficiency

Comparing the relative efficiency of the various approaches is a somewhat delicate exercise, as the results will depend on a number of assumptions which are sometimes poorly stated.

For example, certain curves are easier to work on with BB_1 than with BF: is more or less fair to compare them on identical curves? Also, are we comparing full-fledged encryption, or just KEMs: the latter achieve smaller ciphertexts by deferring some the complexity to the DEM, which is particularly problematic when encrypting for multiple recipients. Do we even take savings brought by multi-recipient encryption into account?

In email and similar one-to-many communication systems (a primary application of IBE), the main efficiency bottlenecks are speed of encryption, speed of private key extraction, and, in mobile applications, size of ciphertext. Decryption speed tends to matter to a lesser extent.

Detailed comparisons will be made in Appendix A. In summary, BB_1 and SK/BB_2/Gentry are very comparable in terms of space and time efficiency. BF is penalized by its comparatively much slower encryption and private key extraction. (Some derivatives of BB_1 such as Waters' also fare poorly in terms of space efficiency, but schemes such as these were not designed for practicality.) The respective KEM versions of these schemes enjoy much shorter ciphertexts than the full versions, though this benefit is negated by the non-reusability of the session key for multiple recipients.

3.4 Implementation Simplicity

The three pairing-based frameworks all have fairly simple algorithmic descriptions — if one had to give a strict ordering, the simplest scheme would be BF, followed by SK/BB_2/Gentry, and then BB_1 and its derivatives. Under the hood, BF hides a unique and potentially non-trivial complication, because it requires the implementation of full domain hashing in the bilinear group (which also has compatibility implications with certain types of curve, as we already noted).

Thus, simplicity is perhaps the one comparison point where the "Exponent Inversion" framework nominally outshines the other approaches. However, the gap with "Commutative Blinding" is too small to be consequential.

4 Mathematical Background

We now very briefly review the notion of cyclic groups equipped with a bilinear map, or pairing.

For concreteness, and for the sake of simplified notation, we shall treat symmetric (type-1) and asymmetric (type-2 and type-3) bilinear maps separately. It will matter little whether an asymmetric pairing comes with (type-2) or without (type-3) an efficient homomorphism $\phi : \hat{\mathbb{G}} \to \mathbb{G}$, since BB_1 neither requires nor forbids the existence of one.

4.1 Symmetric Bilinear Groups

Let $\mathbb{G}$ be a cyclic group of prime order p generated by $g \in \mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ be a function mapping pairs of elements of $\mathbb{G}$ to elements of some group $\mathbb{G}_t$ also of order p. We use a multiplicative notation for the group operations in $\mathbb{G}$ and $\mathbb{G}_t$. To fix ideas, in practice $\mathbb{G}$ will be a subgroup of the group of points on a curve defined over some finite field, while $\mathbb{G}_t$ will be a multiplicative subgroup in some extension of the field. Note that it will not be feasible to compute a homomorphism from $\mathbb{G}_t$ to $\mathbb{G}$ without violating our complexity assumptions.

Suppose that e satisfies the bilinearity condition that $\forall u, v \in \mathbb{G}, \forall a, b \in \mathbb{Z}, e(u^a, v^b) = e(u, v)^{ab}$, and the non-degeneracy condition that $e(g, g)$ generate $\mathbb{G}_t$. Suppose also that the group operations in $\mathbb{G}$ and $\mathbb{G}_t$, as well as the pairing e, can all be computed efficiently (which requires that the elements of $\mathbb{G}$ and $\mathbb{G}_t$ have compact representations). In this case, it is said that $\mathbb{G}$ is a bilinear group, and that the map e is a symmetric bilinear map — or pairing — in the group $\mathbb{G}$. The symmetry refers to the invariance of the bilinear map upon interchange of its arguments.

4.2 Asymmetric Bilinear Groups

Let $\mathbb{G}$ and $\hat{\mathbb{G}}$ be a pair of cyclic groups of prime order p respectively generated by $g \in \mathbb{G}$ and $\hat{g} \in \hat{\mathbb{G}}$. Let $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ be a function mapping pairs of elements in $(\mathbb{G}, \hat{\mathbb{G}})$ to elements of some group $\mathbb{G}_t$ also of order p. Group operations are written multiplicatively in $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$.

Suppose that e satisfies the bilinearity condition that $\forall u \in \mathbb{G}, \forall v \in \hat{\mathbb{G}}, \forall a, b \in \mathbb{Z}, e(u^a, v^b) = e(u, v)^{ab}$, and the non-degeneracy condition that $e(g, \hat{g})$ generate $\mathbb{G}_t$. Suppose also that the group operations in $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$, as well as the pairing e, are all efficiently computable. In this case, it is said that $(\mathbb{G}, \hat{\mathbb{G}})$ form a bilinear group pair, and that the map e is an asymmetric bilinear map, or pairing, in $(\mathbb{G}, \hat{\mathbb{G}})$. The asymmetry refers to the non-interchangeability of the bilinear map's arguments.

Finally, let $\phi : \hat{\mathbb{G}} \to \mathbb{G}$ be the group homomorphism such that $\phi(\hat{g}) = g$; this homomorphism always exists but is not always efficiently computable. When the homomorphism ϕ is efficiently computable, the bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ is of type 2, otherwise it is of type 3. (Remark that when both ϕ and ϕ^{-1} are efficiently computable, it is easy to transform the asymmetric pairing into a symmetric one; in particular, pairings of type 1 are those for which the associated homomorphism is the identity function.)

Realization. Bilinear maps such as the Weil and Tate pairings can be efficiently computed on algebraic curves over finite fields, using an algorithm first proposed by Miller [Mil04]. We refer to the books [BSS99] and the collection [BSS05] for a wealth of information on subsequent improvements.

In this document, since the BB_1 systems are compatible with — and take full advantage of — all known pairings, we shall mostly brush aside the issue of selecting good curves and pairings.

5 Proposed Cryptosystem

We describe the two proposed instantiations of the BB_1 IBE system in the random oracle model. We give two versions: a key encapsulation and a full encryption system. Both versions are fully secure against adaptive chosen identity, chosen ciphertext attacks ($\mathsf{IND\text{-}ID\text{-}CCA}$) in the random oracle model.

Notation. For conciseness, we describe our systems using the asymmetric pairing notation only, using the "hat" ($\hat{}$) diacritic to denote elements of $\hat{\mathbb{G}}$. Thus, to specialize the descriptions to the symmetric case, it suffices to drop all "hats" from the notation (and discard the redundant instance of any duplicate symbol that may appear as a result).

We refer the reader to the full version of [BB04] for definitions, models, proofs, and references.

Security Parameter. The following descriptions do not describe the explict choice of bilinear groups in function of the (implied) security parameter. Rather, we assume that the prime order p, the bilinear groups $\mathbb{G}$ and $\hat{\mathbb{G}}$, the multiplicative group $\mathbb{G}_t$, and the session key space $\{0,1\}^\ell$, are all of size commensurate with the desired level of security. See, *e.g.*, [MOV93] or [Jou04] for details.

5.1 Full Encryption Version

We first describe a (fully secure) full encryption version of BB_1, which allows a sender to encrypt any message of its choice. The message is typically short but need not be, and the same message may be encrypted multiple times for different recipients. In a hybrid system, the message will be an ephemeral session key for bootstrapping a symmetric-key system used to encrypt the actual data.

The recipient identities are represented as distinct but otherwise arbitrary bit strings in $\{0,1\}^*$. The messages (or session keys) to be encrypted are specified as bit strings in $\{0,1\}^\ell$, where the length ℓ can be arbitrarily large. We stress that in this instantiation of BB_1, the originator has full control over the message to be encrypted.

Let g and $\hat{g}$ be the respective generators of some bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ of prime order p, and let $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ be a bilinear map in $(\mathbb{G}, \hat{\mathbb{G}})$. Additionally, we require the availability of three cryptographic hash functions viewed as random oracles:

1. a function $H : \{0,1\}^* \to \mathbb{Z}_p$ for hashing the recipient identity;

2. a function $H' : \mathbb{G}_t \to \{0,1\}^\ell$ for xor-ing with the cleartext; and

3. a function $H'' : \mathbb{G}_t \times \{0,1\}^\ell \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_p$ to make the ciphertext non-malleable.

The BB_1 IBE version works as follows:

***Setup*:** To generate IBE system parameters, first select three integers α, β, and $\gamma \in \mathbb{Z}_p$ at random. Set $g_1 = g^\alpha$, $g_2 = g^\beta$, $g_3 = g^\gamma$ in $\mathbb{G}$, and $\hat{g}_1 = \hat{g}^\alpha$, $\hat{g}_2 = \hat{g}^\beta$, and $\hat{g}_3 = \hat{g}^\gamma$ in $\hat{\mathbb{G}}$. Compute $\hat{g}_0 = \hat{g}^{\alpha\beta}$ and $v_0 = e(g, \hat{g}_0) = e(g, \hat{g})^{\alpha\beta}$. The public system parameters *params* and the master secret key *masterk* are given by:

$$\mathit{params} = (g, g_1, g_3, v_0) \in \mathbb{G}^3 \times \mathbb{G}_t, \qquad \mathit{masterk} = (\hat{g}, \alpha, \beta, \gamma) \in \hat{\mathbb{G}} \times {\mathbb{Z}_p}^3.$$

The generator $\hat{g}$ need not be kept secret, but neither does it need to be published, which is why it is listed as part of *masterk* rather than *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:

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

***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 $c = M \oplus H'(k) \in \{0,1\}^\ell$, calculates $c_0 = g^s$ and $c_1 = g_3^s\, g_1^{H(\mathsf{ID})\, s}$ in $\mathbb{G}$, sets $t = s + H''(k, c, c_0, c_1) \bmod p$, 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 $k = e(c_0, d_0)/e(c_1, d_1) \in \mathbb{G}_t$ and $s = t - H''(k, c, c_0, c_1) \in \mathbb{Z}_p$, and verifies that:

$$(k, c_0) \overset{?}{=} \left(v_0^s, g^s\right).$$

If the componentwise equality does not hold, it rejects the ciphertext; otherwise, it outputs:

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

Variation. Notably, there is an alternate form of the master key that consists only of group elements: $\mathit{masterk}' = (\hat{g}, \hat{g}_0, \hat{g}_1, \hat{g}_3) \in \hat{\mathbb{G}}^4$. From it, private keys with the same distribution as before can be computed using a modified *Extract*′ algorithm: $d_{\mathsf{ID}} = \left(\hat{g}_0\, \hat{g}_3^r\, \hat{g}_1^{H(\mathsf{ID})\, r}, \hat{g}^r\right)$, for random $r \in \mathbb{Z}_p$.

The benefit of the alternate form is that there is no secret integer to be kept, which is beneficial in verifiable secret sharing applications [BBH06]; key extraction is however slightly less efficient.

Performance. The execution speed is markedly faster than the scheme description might suggest, for a couple of reasons.

In *Extract*, *Encrypt*, and *Decrypt*, all group exponentiations consists in raising a fixed base (such as g, g_1, g_3, $\hat{g}$, v_0) to an integer exponent. Notice that the exponentiation bases do not change for the life of the system parameters, and are not tied to any identity. This makes it easy to precompute and permanently store a powers-of-two ladder for each base, which can greatly speed up subsequent exponentiations. With a small store, it is easy to reduce the time of each fixed-base exponentiation to less than 0.2× that of a general exponentation in the same group.

The *Decrypt* algorithm is the only one to require a pairing, besides *Setup*. Although *Decrypt* appears to compute two pairings, their arrangement in a product (or ratio) makes their computation faster than if they had been separate. A recent detailed study [GS06] shows that each additional pairing in a product (or ratio) adds between 0.1× and 0.5× the cost of the first pairing. In summary:

Extract's computational burden is a mere 0.4× that of a single general exponentiation in $\hat{\mathbb{G}}$.

Encrypt's total cost is only 0.6× that of an exponentiation in $\mathbb{G}$, plus 0.2× that of one in $\mathbb{G}_t$.

Decrypt's running time is dominated by the pairing ratio, or about 1.2× that of a lone pairing.

5.2 Key Encapsulation Version

We now describe the (fully secure) IBKEM version of the BB_1 system. The main difference between a KEM and a true encryption system is that in a KEM the originator does not choose the message;

rather, it arises from the randomness used in the ciphertext generation. For this reason, the main use of a KEM (Key Encryption Mechanism) is to transport an ephemeral session key, which is then used to encrypt and authenticate the actual message using a DEM (Data Encryption Mechanism). We refer the reader to the literature for more details on the interaction between KEMs and DEMs.

We assume that user identities are represented as arbitrary bit strings of unspecified length. The session keys generated and transported by the KEM will be bit strings of some fixed length ℓ. Recall that in a KEM the originator typically has little or no direct control over the actual session key.

Let g and $\hat{g}$ be the respective generators of some bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ of prime order p, and let $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ be a bilinear map in $(\mathbb{G}, \hat{\mathbb{G}})$. We require the availability of two cryptographic hash functions viewed as random oracles:

1. a function $H : \{0,1\}^* \to \mathbb{Z}_p$ for hashing the recipient identity;

2. a function $H' : \mathbb{G}_t \to \{0,1\}^\ell$ for deriving a random session key.

The BB_1 IBKEM version works as follows:

***Setup*:** To generate IBKEM system parameters, select three integers α, β, and $\gamma \in \mathbb{Z}_p$ at random. Set $g_1 = g^\alpha$, $g_2 = g^\beta$, $g_3 = g^\gamma$ in $\mathbb{G}$, and $\hat{g}_1 = \hat{g}^\alpha$, $\hat{g}_2 = \hat{g}^\beta$, and $\hat{g}_3 = \hat{g}^\gamma$ in $\hat{\mathbb{G}}$. Compute $\hat{g}_0 = \hat{g}^{\alpha\beta}$ and $v_0 = e(g, \hat{g}_0) = e(g, \hat{g})^{\alpha\beta}$. The public system parameters *params* and the master secret key *masterk* are given by:

$$\mathit{params} = (g, g_1, g_3, v_0) \ \in \mathbb{G}^3 \times \mathbb{G}_t, \qquad \mathit{masterk} = (\hat{g}, \alpha, \beta, \gamma) \ \in \hat{\mathbb{G}} \times {\mathbb{Z}_p}^3.$$

***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:

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

***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:

$$K = H'(v_0^s) \ \in \{0,1\}^\ell, \qquad C = \left(g^s, \ \ g_3^s \, g_1^{H(\mathsf{ID})\, s}\right) \ \in \mathbb{G} \times \mathbb{G}.$$

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:

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

Remarks. The BB_1-IBKEM enjoys the same high performance as the BB_1-IBE system.

The BB_1-IBKEM also achieves chosen ciphertext security (for a KEM) despite the lack of integrity check in *Decapsulate*, due to the intervention of the random oracle H' in the final step of the creation/decryption of the session key K. In a typical KEM/DEM hybrid encryption suite, an integrity check will still be needed for CCA security, but it will be relegated to the DEM level, where it can be implemented using a keyed MAC.

We also mention that, exactly as the BB_1-IBE system, the BB_1-IBKEM admits an alternate form of the master key that comprises only group elements.

6 Practical Extensions

We very briefly mention a few useful extensions of practical significance to the IBE and IBKEM schemes above. These extensions are all straightforward to implement in the BB_1 paradigm.

6.1 Multiple Recipients

Encryption for multiple recipients in the IBE case is as easy as encrypting the given plaintext independently for each recipient, creating one IBE ciphertext for each recipient. The IBE plaintext may be a session key for downstream encryption of an actual message; in that case the downstream ciphertext need not be replicated as long as the same session key is used for each recipient. Hybrid encryption techniques have been well studied and will not be detailed here.

6.1.1 Multi-Recipient IBKEM

The KEM case is a bit more complicated. In a KEM the session key is not under the control of the sender, and cannot be directly replicated from one recipient to the next. It is necessary to interpose a supplemental layer to create a intermediate session key that remains the same for all recipients. In the random oracle model, this can be done with the help of a supplemental hash function (in addition to the function H and H' described in Section 5.2):

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

Multi-Recipient BB_1-IBKEM is obtained by modifying the above BB_1-IBKEM as follows:

***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 a random $s_i \in \mathbb{Z}_p$ for each recipient, and outputs:

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

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:

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

6.2 Streaming

Streaming encryption and/or decryption are desirable whenever the size of the data is not known in advance, or is too large to allow buffering. This is usually achieved using a hybrid KEM/DEM scheme, where the KEM header conveys the session key for a DEM that supports streaming operation. Clearly, the same technique can be used with the BB_1 IBKEM of Section 5.2.

By contrast to many CCA-secure monolithic encryption systems, the full BB_1-IBE of Section 5.1 can also operate in streaming mode, both during encryption and decryption, for messages of any length in $\{0,1\}^*$. To achieve this, we need the function H' to produce a streaming output in $\{0,1\}^*$ for xor-ing with M given as a stream, and the function H'' to be incrementally computable for any stream argument $c \in \{0,1\}^*$. The ciphertext then needs to be reordered so that c_0 and c_1 appear first, followed by the string or stream c; the checksum t must come last.

We note that in order for an application of streaming decryption to provide CCA security, it must allow the recall (and erasure) of all decryption output *ex post facto* in case of decryption failure (*i.e.*, failure of the integrity test). Similar restrictions also apply to the KEM/DEM framework.

6.3 Secret Sharing

IBE secret sharing refers the ability to split the master key, *masterk*, into distinct shares, to prevent abuse by a single trusted authority. In a sharing scheme, the trusted authority is replaced by a number of partial authorities, that have the power to map identities to partial private keys only.

In a t-out-of-n sharing scheme, a user will need to obtain t valid partial private keys from t distinct authorities (subject to the same system parameters), in order to reconstruct his or her full private key — without which decryption cannot be done.

Our systems based on the BB_1 method can easily be modified to support threshold secret sharing using Lagrange polynomial interpolation. The details may be found in [BBH06].

6.4 Hierarchical Identities

It is often useful to allow identities to be hierarchical, be it to replicate an organizational hierarchy, or for individual users to create persona with restricted capabilities. In a hierarchical IBE, or HIBE, identities take the form $\mathsf{ID} = a.b.c.d$. This particular identity lives at depth 4, and is subordinate to a, $a.b$, and $a.b.c$, but not to $a.e$. Each user in the hierarchy may act as a local authority for all subordinate identities.

The BB_1 system is well suited for efficient hierarchical operation; see [BB04] for details. In particular, Boneh, Boyen, and Goh [BBG05] describe a very efficient and compact HIBE based on BB_1, and that readily applies to the instantiations of BB_1 given in Section 5.

6.5 Forward Security

Recall that it is customary in IBE deployments to let the ciphertexts and private keys depend upon a timestamp, which is incremented at regular intervals (such as once a week). The purpose is to make private keys sufficiently short lived to reduce or eliminate the need for revocation lists.

Forward security, in the context of IBE, allows the trusted authorities to update their master keys in a non-reversible manner according to a similar schedule. Forward security protects the privacy of any ciphertext created before an update, even in the case of exposure of the current master key.

Forward secure IBE (fs-IBE) can be achieved generically by using short-lived "throw-away" master keys and corresponding public parameters; this is however very impractical. Canetti, Halevi, and Katz [CHK03] show to construct fs-IBE with a single set of public parameters, using time hierarchies. Very efficient fs-IBE and fs-HIBE extensions to BB_1 are described in [BBG05].

6.6 Recipient Anonymity

Anomymity in the context of public-key or identity-based encryption is the ability to keep the intended recipient of a ciphertext a secret (except from the recipient him- or herself). Sender anonymity is also desirable, but typically achieved automatically unless there is an authentication mechanism that acts on the ciphertext.

Anonymous IBE has a special appeal that nicely complements the inherent safeguards that IBE provides against traffic analysis: unlike the classic Public Key Infrastructure (PKI) model, IBE does not require a sender to look up the certificate of a recipient in order to engage in communication.

The BF system is natively anonymous. The BB_2 and SK systems are not (although one should note that the Gentry-IBE system is). The BB_1 approach does not provide anonymity by default, but it can be modified appropriately to support it — even in the hierarchical case, which is another unique property afforded by BB_1's Commutative Blinding IBE paradigm.

7 Concluding Remarks

The purpose of this note was to give a standard description of the BB_1 identity-based cryptosystem. Two "fully secure" instantiations have been presented: one as an encryption system, the other as a key encapsulation mechanism. Our systems only make generic use of the group and pairing operations, and can be implemented without impediment or performance penalty on any algebraic curve that supports a pairing.

Additionally, we have made the case for BB_1 as a method of choice for all applications that require an identity-based functionality. Our detailed study shows that the BB_1 system sustains the comparison with every other identity-based system in almost every respect. The BB_1 paradigm offers an excellent blend of raw efficiency, ease of implementation, extensibility, and broad utility.

References

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–38. Springer-Verlag, 2004. http://www.cs.stanford.edu/~xb/eurocrypt04b/.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–56. Springer-Verlag, 2005. http://www.cs.stanford.edu/~xb/eurocrypt05a/.

[BBH06] Dan Boneh, Xavier Boyen, and Shai Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In *Proceedings of RSA-CT 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 226–43. Springer-Verlag, 2006. http://www.cs.stanford.edu/~xb/ctrsa06/.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–29. Springer-Verlag, 2001.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *Proceedings of Asiacrypt 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–32. Springer-Verlag, 2001.

[BMW05] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security—CCS 2005*. ACM Press, 2005. http://www.cs.stanford.edu/~xb/ccs05/.

[BSS99] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic Curves in Cryptography*, volume 265 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1999.

[BSS05] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart, editors. *Advances in Elliptic Curve Cryptography*, volume 317 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 2005.

[BW06] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Advances in Cryptology—CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006. `http://www.cs.stanford.edu/~xb/crypto06a/`.

[CCMLS05] Liqun Chen, Zhaohui Cheng, John Malone-Lee, and Nigel P. Smart. An efficient ID-KEM based on the Sakai-Kasahara key construction. Cryptology ePrint Archive, Report 2005/224, 2005. `http://eprint.iacr.org/2005/224/`.

[Che06] Jung Hee Cheon. Security analysis of the Strong Diffie-Hellman problem. In *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, 2006.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, 2001.

[CS05] Sanjit Chatterjee and Palash Sarkar. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In *Proceedings of ICISC 2005*, 2005.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2006*, Lecture Notes in Computer Science. Springer-Verlag, 2006.

[GPS06] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. `http://eprint.iacr.org/2006/165/`.

[GS06] Robert Granger and Nigel P. Smart. On computing products of pairings. Cryptology ePrint Archive, Report 2006/172, 2006. `http://eprint.iacr.org/2006/172/`.

[Jou04] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–76, 2004. Extended abstract in *Proceedings of ANTS IV, 2000.*

[Mil04] Victor Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4), 2004.

[MNT01] Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals*, E84-A(5):1234–43, 2001.

[MOV93] Alfred Menezes, Tatsuaki Okamoto, and Scott Vanstone. Reducing elliptic curve logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–46, 1993.

[MSK02] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A new traitor tracing. *IEICE Transactions on Fundamentals*, E85-A(2):481–4, 2002.

[Nac05] David Naccache. Secure and practical identity-based encryption. Cryptology ePrint Archive, Report 2005/369, 2005. `http://eprint.iacr.org/2005/369/`.

[SK03] Ryuichi Sakai and Masao Kasahara. ID based cryptosystems with pairing over elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003. `http://eprint.iacr.org/2003/054/`.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science.* Springer-Verlag, 2005.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science.* Springer-Verlag, 2005.

A Detailed Comparisons

We now set our sight on a factual comparison between the two proposed instantiations of BB_1 and the respectively competing IBE and IBKEM approaches. These comparisons serve to support and supplement the general arguments already given in Sections 2 and 3.

A.1 Compared Systems

Below is a list of identity-based schemes to be compared. To avoid comparing apples and oranges, we must separate the IBE from the IBKEM instantiations. To ensure a fair a valid comparison, we attempt to give data for both versions whenever supported by a given scheme.

The main functional difference between IBE and IBKEM is that the former can carry a message chosen by the sender, and hence requires a correspondingly longer ciphertext (thus, one should compare the ciphertext *overhead*, which discounts the size of the message, if any). On the contrary, it is usually not possible, or not safe, to encapsulate the same session key more than once in an IBKEM, without interposing a third layer between the basic KEM and the DEM. Consequently:

An IBKEM/DEM combination is well suited for single-recipient encryption.

An IBE coupled with a DEM will be more practical for multiple recipients.

A.1.1 IBE Schemes

Our list of IBE systems to be compared is as follows:

1. BB_1**-IBE.** This is the first Boneh-Boyen scheme [BB04, §4], instantiated with identity and ciphertext hashing as described in Section 5.1. It has a full $\mathsf{IND\text{-}ID\text{-}CCA}$ proof of security in the random oracle model. It is a "commutative blinding" scheme per our classification.

2. BB_2**-IBE.** This is a simple adaptation of the second Boneh-Boyen scheme [BB04, §5], with identity and ciphertext hashing in the random oracle model to achieve the full notion of security as efficiently as possible. We do not describe the scheme in detail (though see below for a KEM variant called SK-IBKEM). It uses a strong assumption, and is an "exponent inversion" scheme per our classification.

3. **BF-IBE.** This is the original Boneh-Franklin scheme with Fujisaki-Okamoto padding for full security in the random oracle model, as described in [BF01]. It is a "full domain hash" scheme per our classification.

For the sake of completeness, we also mention the following IBE schemes, which will not be compared. Although all of these schemes have theoretical appeal, none of them brings practical value over the IBE systems mentioned above.

- BB_1-IBE Without Random Oracles [BB04, §4]. This is the plain version of the Boneh-Boyen BB_1 scheme from 2004, which either satisfies only a weaker notion of security, or requires a significantly larger group size to achieve the same level of security as the random oracle version described in this document.

- Waters-IBE [Wat05] and its improvements [CS05] and [Nac05]. These are variants of the BB_1-IBE scheme with better proofs of security without random oracles, but at the cost of lesser efficiency and impractically large public system parameters.

- Gentry-IBE [Gen06]. This scheme is superficially very similar to BB_2/SK-IBE, though with a crucial difference that gives it a tight proof of security without random oracles. Nevertheless, the Gentry scheme is less efficient than random oracle versions of BB_2/SK-IBE, and requires an even stronger complexity assumption (of the "large q" variety; see Section 3.1).

A.1.2 IBKEM Schemes

The list of IBKEM systems we compare is as follows:

1. **BB_1-IBKEM.** This is the BB_1 scheme [BB04, §4] instantiated as an identity-based KEM in the random oracle model, as described in Section 5.2. It is a "commutative blinding" scheme per our classification.

2. **SK-IBKEM.** This is a modern variant of the Sakai-Kasahara scheme [SK03], adapted to allow a very similar security proof as the second Boneh-Boyen scheme, albeit in the random oracle model. Since this KEM scheme has been described in detail by Chen *et al.* [CCMLS05], this is the version we will use in our comparison. It is an "exponent inversion" scheme per our classification.

3. **BF-IBKEM.** This is a KEM adaptation of the Boneh-Franklin scheme [BF01], in the random oracle model. We do not give an explicit description of this scheme; we mention it merely because it provides an interesting comparison benchmark. It is a "full domain hash" scheme per our classification.

For completeness, we mention the following IBKEM system, which will not be compared.

- BMW-IBKEM [BMW05]. This is a KEM built upon a two-level hierarchical (2-HIBE) instantiation of the BB_1 system. It is one of the most efficient such systems to achieve chosen ciphertext security without random oracles, but it is still not as efficient as BB_1-IBKEM in the random oracle model.

A.2 Comparison Data

The following table provide extensive comparison data for the IBE and IBKEM systems listed above. We use the following symbols to qualify certain Yes/No comparisons:

YES:

(✓) the checkmark indicates the presence of a feature that is desirable;

($\sim$) the tilde denotes the presence of an ambivalent characteristic;

($\times$) the cross points out a negative or unfortunate characteristic;

NO:

( ) an empty space signals the *absence* of a characteristic.

A.2.1 Security

The following table shows that all systems satisfy the strongest security requirement of IND-ID-CCA security, albeit in the random oracle model. The reliance on random oracles is not uniform, and we note some general trends:

- "Commutative blinding" systems (BB_1) are the most amenable to random oracle elimination. These systems can make good use of random oracles; nevertheless, they can achieve all the usual security properties in the standard model.
- "Exponent inversion" schemes (BB_2 and SK) generally require random oracles for efficient chosen ciphertext security, since they lack the flexibility needed for many of the cleverer tricks. However, this is not a universal rule; for instance, Gentry-IBE [Gen06] achieves CCA security with reasonable efficiently using the venerable method of double encryption.
- "Full domain hash" systems (BF) are highly dependent on random oracles for essentially all of their security properties. No provable security remains outside of the random oracle model for these schemes.

Aside from random oracles, the main security distinction between the various systems lies in the widely differing strength of the assumptions they require. BDH and q-BDHI for very small values of q can be viewed as mild assumptions in bilinear groups. However, as discussed in Section 3.1, q-BDHI for very large values of q is a more worrisome assumption to rely upon, as evidenced by a recent attack on "large q"-BDHI and related assumptions, by Cheon [Che06].

	Full IBE			IBKEM Only		
	BB_1-IBE	BB_2-IBE	BF-IBE	BB_1-KEM	SK-KEM	BF-KEM
Provable Security						
IND-ID-CCA	✓(RO)	✓(RO)	✓(RO)	✓(RO)	✓(RO)	✓(RO)
proof with RO	×	×	×	×	×	×
and without RO?	sID-CCA*	sID-CPA	void	sID-CCA*	void	void
Assumption Strength						
q-BDHI (large q)		×			×	
q-BDHI (small q)	∼**			∼**		
BDH (weakest)	✓		✓	✓		✓

* full IND-ID-CCA possible without RO by enlarging the groups [BB04, §7] or the system parameters [Wat05]

** option to use q-BDHI instead of BDH for improved efficiency in BB_1-like hierarchical systems of depth q [BBG05]

Cheon's Attack

Cheon's [Che06] attack does not spell doom nor does it shatter the security of "exponent inversion" schemes such as BB_2, SK, and Gentry's, but it reduces their real-world efficiency, by making it necessary to increase the size of the groups (by up to 50% more bits) for any given security level. In all the tables that follow, we shall ignore the ramifications of Cheon's attack, and leave it to the reader to apply the appropriate penalties.

A.2.2 Compatibility

The following table shows the types of pairings that can be used to implement each system.

The BF systems demand the ability to hash.

The BB_2/SK schemes necessitate a homomorphism in the security reduction.

The BB_1 instantiations are compatible with all known pairings.

	Full IBE			IBKEM Only		
	BB_1-IBE	BB_2-IBE	BF-IBE	BB_1-KEM	SK-KEM	BF-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 $\mathbb{G}_t$						

* subject to feasibility of full domain hash into $\hat{\mathbb{G}}$

A.2.3 Versatility

The following table 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. The "exponent inversion" schemes BB_2 and SK are severely limited.

	Full IBE			IBKEM Only		
	BB_1-IBE	BB_2-IBE	BF-IBE	BB_1-KEM	SK-KEM	BF-KEM
Multi-Recipient	✓	✓	✓	$\sim^{*}$	$\sim^{**}$	
Streaming	✓			✓	✓	✓
Master-Key Sharing						
n-out-of-n	✓		✓	✓		✓
t-out-of-n	✓		✓	✓		✓
Hierarchical Identities						
linear-size CT	✓		✓	✓		✓
const.-size CT	✓			✓		
Forward Security						
list (lin-size)	(✓)	×	(✓)	(✓)	×	(✓)
tree (log-size)	✓		✓	✓		✓
compr'd (< log)	✓			✓		
Recipient Anonymity						
regular	$\sim^{***}$		✓	$\sim^{***}$		✓
hierarchical	$\sim^{***}$			$\sim^{***}$		

* with additional intermediate session key layer (see Section 6.1.1)

** with built-in intermediate session key layer (see [CCMLS05])

*** full anonymity for arbitrary hierarchies requires extension to the basic scheme [BW06]

A.2.4 Space Efficiency

The following table compares the bandwidth (or memory) overheads associated with each scheme.

The ciphertext overhead for encryption specifically exclude the number of message bits that can be freely chosen by the sender; the actual ciphertext size is equal to the ciphertext overhead plus the message size. For KEM, the message size is considered to be zero, since it cannot be chosen by the sender; the entire capsule is thus counted as overhead.

Even with the above correction, one should exercise caution when comparing IBE ciphertext overheads with IBKEM encapsulation overheads: the reason being that a KEM is useless by itself and must be adjoined a "sufficiently secure" DEM in order to become useful. The DEM module will incur an additional overhead on the entire system, that has no mandatory counterpart in a self-contained encryption scheme.

The overheads for the various constructions are listed in the table below. For the sake of completeness, we have also listed the master key sizes, even though these are for all practical purposes essentially irrelevant. The same is true of private key sizes, to a lesser extent.

	Full IBE			IBKEM Only		
	BB_1-IBE	BB_2-IBE	BF-IBE	BB_1-KEM	SK-KEM	BF-KEM
System Parameters						
# elts. $\in \mathbb{G}$	3	2	2	3	2	2
# elts. $\in \mathbb{G}_t$	1	1	0	1	1	0
Master Secret						
# elts. $\in \mathbb{Z}_p$	3	1	1	3	1	1
# elts. $\in \hat{\mathbb{G}}$	1	0	0	1	0	0
Private Key						
# elts. $\in \hat{\mathbb{G}}$	2	1	1	2	1	1
Ciphertext*						
# elts. $\in \mathbb{Z}_p$	1	0	0			
# elts. $\in \mathbb{G}$	2	1	1			
extra overhead	–	n bits**	n bits**			
KEM Capsule						
# elts. $\in \mathbb{Z}_p$				0	0	0
# elts. $\in \mathbb{G}$				2	1	1
extra overhead				–	n bits**	–

* the ciphertext overhead for encryption excludes the size of the message

** extra overhead caused by random oracle output of size $n \geq 2\sigma$ bits at σ-bit security level

A.2.5 Time Efficiency

The following table 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.

To facilitate comparisons, each type of operation within a given algebraic group is assigned a coefficient indicating its cost relative to a general exponentiation in the same group. These coefficients are fairly independent of the group under consideration, as long as comparisons are confined to the same group. As a general rule, the factors mostly affecting computational costs are:

1. the number of *independent* pairings (products and ratios of pairings count as one);

2. to a lesser extent, the number of *general* exponentiations and hashing operations.

Due to their much higher dependence on the curve selection, relative coefficients across groups are not listed in this table. However, specific examples will be given in Section A.3.

		Full IBE			IBKEM Only		
		BB_1-IBE	BB_2-IBE	BF-IBE	BB_1-KEM	SK-KEM	BF-KEM
Private Key Extraction							
# hash to grp.	(1×)			$\hat{\mathbb{G}}$			$\hat{\mathbb{G}}$
# general exp.	(1×)			$\hat{\mathbb{G}}$			$\hat{\mathbb{G}}$
# fix-base exp.	(0.2×)	$\hat{\mathbb{G}}\ \hat{\mathbb{G}}$	$\hat{\mathbb{G}}$		$\hat{\mathbb{G}}\ \hat{\mathbb{G}}$	$\hat{\mathbb{G}}$	
Encryption/ Encapsulation							
# pairing	(5×)			$\rightarrow \mathbb{G}_t$			$\rightarrow \mathbb{G}_t$
# hash to grp.	(1×)			$\hat{\mathbb{G}}$			$\hat{\mathbb{G}}$
# general exp.	(1×)			$\mathbb{G}_t$		$\mathbb{G}$	$\mathbb{G}_t$
# fix-base exp.	(0.2×)	$\mathbb{G}\ \mathbb{G}\ \mathbb{G}\ \hat{\mathbb{G}}_t$	$\mathbb{G}\ \mathbb{G}\ \hat{\mathbb{G}}_t$	$\mathbb{G}$	$\mathbb{G}\ \mathbb{G}\ \mathbb{G}\ \hat{\mathbb{G}}_t$	$\mathbb{G}\ \hat{\mathbb{G}}_t$	$\mathbb{G}$
Decryption/ Decapsulation							
# pairing ratio	(6×)	$\rightarrow \mathbb{G}_t$			$\rightarrow \mathbb{G}_t$		
# pairing	(5×)		$\rightarrow \mathbb{G}_t$	$\rightarrow \mathbb{G}_t$		$\rightarrow \mathbb{G}_t$	$\rightarrow \mathbb{G}_t$
# general exp.	(1×)					$\mathbb{G}$	
# fix-base exp.	(0.2×)	$\mathbb{G}\ \hat{\mathbb{G}}_t$	$\mathbb{G}\ \mathbb{G}$	$\mathbb{G}$	–	$\mathbb{G}$	$\mathbb{G}$

A.3 Illustrative Comparison with Concrete Parameters

To illustrate what the above tables mean in practice, we restate our time and space comparisons in the concrete case of type-1 pairings on supersingular (SS) curves of large characteristic used in [BF01], and type-2 pairings on MNT curves [MNT01] as described in [BLS01]. We consider the 80-bit and 128-bit security levels, which respectively require elliptic curve group sizes of 160 and 256 bits to thwart generic attacks, and finite fields of 1024 and 3072 bits to defeat the number field sieve. Depending on the choice of curve, it may not be possible to satisfy both constraints tightly, and thus the actual representations will typically need to be larger.

Since SS and MNT curves have embedding degree 2 and 6, we work out the minimum representation sizes to be, for the chosen security levels:

	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

Similarly, we estimate the approximate relative costs of the various group and pairing operations to be (after normalization to the cost of a general exponentiation in $\mathbb{G}$, for each curve):

	Relative Timings (1 unit = general expon. in $\mathbb{G}$)*		
	SS @ 80-bit security	MNT @ 80-bit security	MNT @ 128-bit security
In $\mathbb{G}$:			
fix-base expon.	0.2	0.2	0.2
general expon.	1	1	1
In $\hat{\mathbb{G}}$:			
fix-base expon.	0.2	7	7
general expon.	1	36	36
hashing	1	36	36
In/to $\mathbb{G}_t$:			
fix-base expon.	0.8	7	7
general expon.	4	36	36
single pairing	20	150	150
ratio of pairings	24	180	180

* for random exponent in $\mathbb{Z}_p$

For illustration purposes, we can work out the representation overheads and computational costs incurred by the various systems for the three particular choices of curves:

Estimated Values	Full IBE			IBKEM Only		
$\boxed{\text{Space}}\big/\text{Time}$	BB$_1$-IBE	BB$_2$-IBE**	BF-IBE	BB$_1$-KEM	SK-KEM**	BF-KEM
SS @ 80-bit security level						
Overhead (bits)						
public params.	2560	2048	1024	2560	2048	1024
CT (excl. msg.)	1184	672	672			
KEM capsule				1024	672	512
Cost (rel. time)*						
key extraction	< 1	< 1	2	< 1	< 1	2
encryption	2	2	21			
decryption	24	20	20			
encapsulation				2	2	21
decapsulation				24	20	20
MNT @ 80-bit security level						
Overhead (bits)						
public params.	1539	1368	342	1539	1368	342
CT (excl. msg.)	502	331	331			
KEM capsule				342	331	171
Cost (rel. time)*						
key extraction	14	7	72	14	7	72
encryption	8	8	322			
decryption	187	151	150			
encapsulation				8	8	322
decapsulation				180	186	150
MNT @ 128-bit security level						
Overhead (bits)						
public params.	4608	4096	1024	4608	4096	1024
CT (excl. msg.)	1280	768	768			
KEM capsule				1024	768	512
Cost (rel. time)*						
key extraction	14	7	72	14	7	72
encryption	8	8	322			
decryption	187	151	150			
encapsulation				8	8	322
decapsulation				180	186	150

* relative cost unit = time to raise random element in $\mathbb{G}$ to random exponent in $\mathbb{Z}_p$ (for stated curve/security)

** BB$_2$-IBE and SK-KEM group sizes before any potential enlargement to compensate for Cheon's [Che06] attack