# Identity-Based Encryption with Conventional Public-Key Infrastructure

Jon Callas
PGP Corporation
Palo Alto, California, USA
jon@pgp.com

18th February 2005

**Abstract**

This paper proposes an identity-based encryption (IBE) scheme based on traditional public-key cryptographic systems, such as RSA, DSA, Elgamal, etc. This scheme has a number of advantages over other systems. It can rely upon these traditional systems for its security. Since it uses these traditional encryption schemes, it is interoperable with and easily embedded within an existing security system that uses these functions. Additionally, its construction as an on-line system avoids the operational security flaws of IBE systems that allow off-line key generation.

## 1 Introduction

Conceptually, public keys behave a lot like telephone numbers — if I want to call you, I need your telephone number. I don't need my own number to make calls (I can use a pay phone, for example), I need one to receive them. In a fashion that is analogous to a telephone number, I need *your* public key to encrypt something so that it is secret to you.

Unlike telephone numbers, public keys are far too big for people to remember. Even elliptic curve keys, which are much shorter than the traditional ones are far too large for a person to remember. George Miller's classic research [MILLER56] done on telephone numbers is that the average person can remember seven give or take two digits. A 160-bit key will be something over 40 digits long (exactly 40 if we use hexadecimal). So memorizing someone's key the way you memorize their phone number is completely out of the question.

Consequently, we need to have some blobs of data that say that a name such as `alice@example.com` belongs to some key. There is also value in digitally signing the blob so that the receiver has some assurance that the association is accurate. These blobs are *certificates*.

Like any data management problem, certificate management is harder than people would like. This is why in 1984 Adi Shamir suggested the idea of coming up with a crypto scheme in which any string can be a public key [SHAMIR84]. Thus, there is no need to associate a name with a public key, because they're effectively the same. This is *Identity-Based Encryption*.

While typically we think of IBE systems converting names into public keys, it should be possible to make any arbitrary bit-string, $\mathsf{ID} \in \{0,1\}*$, a determinant of a public key in an IBE system. Thus, a name, an email address, or even a binary object such as a picture or sound can be considered equivalent to a public key. Thus, an IBE system can be thought of as a function of the form $K_i = IBE(\mathsf{ID}_i)$ that produces keys from arbitrary bit strings that we call identities, without loss of generality.

## 2 Overview

IBE can also be thought of as an *Attribute-Based Enrollment* mechanism. Its goal is to reduce the overhead required to bring an entity into the system. Thus, we take some attribute of the entity and use that as a functional equivalent to a public key.

In the past, work on IBE has been mathematical. It has developed a new public-key cryptosystem that has as a part of its key creation some arbitrary bitstring that is the identity. We examine this past work and look at how they are put together, as well as the limitations on these previous systems.

Next, we construct a framework for an IBE system that satisfies the basic goals of IBE — that this attribute of an entity, its so-called identity is equivalent to a public key — and uses a parallel structure. However, this new framework can construct key pairs that are of a familiar cryptosystem such as RSA, rather than requiring its users to adopt a new public key algorithm.

This construction also differs from present IBE systems in that it does not allow off-line generation of keys, but we also note that off-line generation has security drawbacks as well as advantages. However, on-line generation also permits a hybrid PKI that has both traditional and identity-based aspects in the same infrastructure.

Lastly, we look at open and unsolved problems that surround IBE systems in general, including this one. All IBE systems created so far have a set of limitations as well as characteristics that are not yet solved. They do not remove the utility or desirability of IBE, but do limit where it can be effectively deployed.

## 3 Components of IBE

An IBE system contains four basic components in its construction:

1. **System Setup**: IBE systems rely upon a trusted central authority that manages the parameters with which keys are created. This authority is called the *Private Key Generator* or PKG. The PKG creates its parameters, including a master secret $K_{pkg}$ from which private keys are created.

2. **Encryption**: When Bob wishes to encrypt a message to Alice, he encrypts the message to her by computing or obtaining the public key, $P_{Alice}$, and then encrypting a plaintext message $M$ with $P_{Alice}$ to obtain ciphertext $C$.

3. **Key Extraction**: When Alice wishes to decrypt the message $C$ that was encrypted to her name, she authenticates herself to the PKG and obtains the secret key $S_{Alice}$ that she uses to decrypt messages.

4. **Decryption**: When Alice has $C$ and $S_{Alice}$, she decrypts $C$ to obtain the plaintext message $M$.

No matter the specific parameters or requirements of the system, these functional aspects are always present in IBE systems as their defining components.

## 4 Previous Work

Shamir's original system was based upon RSA encryption and is a signature-only system. Shamir was unable to extend it to an encryption system. Between 1984 and 2001, a number of IBE systems were created, but they all had limitations, such as requiring that users of the system not collude, or requiring large amounts of computation on the part of the PKG. In 2001, two new proposals were published, improving on previous work.

Clifford Cocks created a scheme based upon quadratic residues [COCKS01]. Cocks's system encrypts bit-by-bit, and requires expansion of the message; for a 1024-bit modulus and a 128-bit bulk encryption key, 16K of data must be transfered. With modern networks, this is a completely acceptable overhead[1].

Dan Boneh and Matt Franklin created a scheme based upon Weil Pairings [BF01]. Pairing-based systems use bilinear maps between groups to establish a relationship whereby hashes of the identity create the encryption scheme. Boneh-Franklin IBE has had further work [BB04] and is an active area of research.

Horwitz and Lynn [HL02], Gentry and Silverberg [GS02] improved upon performance characteristics of a Boneh-Franklin PKG by extending IBE systems to *Hierarchical IBE* (HIBE). Their work is important particularly because of its attention to the practical details of constructing a scalable PKG. Gentry also described *Certificate-Based Encryption* (CBE) that uses an IBE system with certificates to create a hybrid approach [GENTRY03] that essentially makes the "identity" not be a name, but a well-defined certificate. In a conceptually related approach, Al-Riyami and Paterson have their *Certificateless Public Key Cryptography* [AY03].

Benoît Libert and Jean-Jacques Quisquater also created an identity-based signcryption scheme based on pairings [LQ03]. These signcryption schemes combine both aspects into one operation. There is other somewhat related work on combining signing and encryption as well such as [ZHENG97].

[1] Other discussions of IBE have characterized this expansion as an *unacceptable* overhead. Debating how much expansion is tolerable is orthogonal to this paper, but I feel it necessary to explicitly state that I find acceptable what previous authors find unacceptable. Networks continually get faster. Many messages are small enough that other overhead they already have to deal with (like conversion to HTML) also expand them. In the case where the message is large, clever software engineering could use compression and efficient bulk encryption to make this no worse than other message bloat.

# 5 Limitations on Previous Work

All of the existing IBE systems have their own limitations. Shamir's system signed but did not encrypt. Cocks's system needs care to avoid an adaptive chosen ciphertext attack. It is also inefficient, but still efficient enough for use on reasonable communications paths. While others have proofs of security, there is a notoriously poor relationship between proofs of security and actual system security. Security proofs can show where a system is safe, but not protect against new assumptions that an adversary can bring to bear against the system nor against uses of a system that its creators did not think of which may be outside of the scope of the original threat model. Still other subtle problems have shown up on other systems, such as the ability in early HIBE systems for colluding users to determine the PKG's master key.

With the exception of Shamir's system, IBE systems rely on new public-key cryptosystems, most often Weil pairing. Consequently, they are not compatible with existing systems that use RSA, Elgamal, or DSA. This limits their practical application, since there are many existing systems built upon these cryptosystems. Also, experience and comfort with the security of these established systems is high. A key advantage that Shamir's system has over all those that follow it is that it was based on established public key cryptography, and thus (had it been successful in being both a signing and encrypting system) interoperable with non-IBE systems. Had Shamir's system been successful at encrypting, an RSA-based IBE system would likely be the dominant IBE system today, if for no other reason than its interoperability with deployed systems.

This is an important observation — if we can construct an IBE system that uses traditional, integer-based, public key cryptography, the barriers to adoption of IBE systems might be lowered. The value that IBE has can be fully realized if it can be made to work with these established systems. Furthermore, this system has the advantage that it can rely on twenty years of mathematical and operational familiarity with these traditional public-key cryptosystems.

# 6 Security Parameters of the Off-Line and On-Line worlds

Previous IBE systems have as a desirable property that they support off-line generation of keys. That is to say, Bob receives key-generation parameters from the PKG once, and then can generate an arbitrary number of public keys.

While off-line key generation is desirable, it is not without its own security consequences.

## 6.1 Advantages of Off-Line Generation

Off-line generation is ideal in an off-line environment. If communication with the PKG is slow, expensive, or unreliable, then off-line generation is a huge advantage to its users. They need only one interaction with a given PKG to be able to do all needed work with that server.

This advantage becomes less, however, as communication with a PKG becomes cheaper, easier, and faster. One some level, off-line key generation is nothing more than a key server that is an algorithm instead of a database. This is an advantage when databases are static and expensive, but not when databases are cheap and fast. In an environment where the *contents* of the database

are dynamically changing, a database change is not only an algorithm change, but an algorithm change that must be propagated to all clients of the PKG.

## 6.2 Disadvantages of Off-Line Generation

Oftentimes, the strengths of a system are also its weaknesses. This is also true with off-line generation. Off-line generation makes key generation easy not only for legitimate users of of the system but for illegitimate ones.

An issue that PKIs must consider in their design is that of a *Directory Harvest Attack*, in which senders of unwanted advertisements or outright fraudulent confidence games use the directory as a way to discover information paths into the system. Off-line generation of keys allows spammers and other attackers to to pre-generate email attacks in their own system or create a distributed system for encrypted attacks. These attacks are not an issue in off-line systems.

Off-line generation has the disadvantage that there is complete transparency in the directory, since the directory is an algorithm. Anyone with that algorithm has all possible entries in the directory and their public keys, and this can be exploited in side-channel attacks that are not attacks on the cryptographic system *per se*, but the way the system is used.

Off-line generation has as an additional disadvantage increased revocation problems. A conventional PKI must be able to re-issue certificates and handle for revisions in the PKI. An off-line IBE system must not only handle revocation of the certificates themselves but a revocation of the *algorithmic parameters* that comprise its own PKI. No IBE system before this one has even considered this real-world problem.

In fact, the key advantages of this on-line system are that it considers and solves these real-world problems.

## 6.3 On-Line IBE for the On-Line World

Sadly, trends in the real world make the advantages of off-line IBE moot, and turns its disadvantages into outright security problems. There is little need for off-line generation in an on-line world, and the advantages of off-line generation benefit attackers more than defenders.

Nonetheless, IBE has desirable characteristics. The core IBE concept, that there is an equivalence relationship between bit-strings and keys has appeal. Designing an IBE system that has the advantages of name-to-key mapping without the security flaws of off-line key generation can make IBE acceptable to the complex security requirements of the Internet.

Furthermore, if we shift the IBE system to an on-line system, we can construct it so that it uses traditional keys. This permits an IBE system to be embedded within an existing cryptosystem and interoperable with existing systems that use these keys. Not only does this remove adoption issues, but it also simplifies proofs of security; it is trivial to prove that an encryption portion of an IBE system is as secure as RSA if the underlying encryption is RSA.

Another advantage is that an on-line system can normalize the identity. It is common for users of an email system to have equivalent identities on the system. For example `alice@example.com`

and `asmith@example.com` may be the same user, and it is desirable to have only one key. An on-line system can canonicalize identities at runtime.

Finally, and perhaps counterintuituvely, this permits IBE keys to be used in certificates. We usually think of IBE as a way to eliminate certificates. However, all keys require standard data structures for transport. Whatever flaws they have, certificates are existing, standard ways to format key material in a way that systems can reliably use them. Objections to certificate-based systems are not objections to the *certificates* per se, but to the *certification process*. Without a standard set of transport data structures, IBE proponents must standardize on key transport data structures and convince developers to use those structures as well as the new crypto algorithms and protocols. Using existing certificate systems reduces the Key Extraction problem to an existing problem that has a simple solution, e.g. a lookup in a directory.

Combining certificates with IBE is not new to this proposal. Gentry's CBE combines a form of certificates with Weil pairings.

On-line systems are ubiquitous and becoming more available every day. Consequently, the advantage of off-line key generation in an IBE system not only has less value today than it did when Shamir first suggested IBE in 1984, but new attacks turn it into a boon for the attacker of a system. Relaxing the parameters of an IBE system so that Bob is required to ask the PKG for each key is certainly practical, and permits us to exploit these other desirable system features.

# 7 Constructing IBE to Use Conventional Cryptography

It is a goal of this system to describe how to construct an IBE from well-known components that have easily-understood security constraints, including proofs of security. Thus, what follows is actually a *adaptive framework* for constructing an IBE system that is not bound to a single algorithm and is functional even in the face of security advances such as new attacks on hash functions [BIHAMCHEN04] [JOUX04] [WANG04].

## 7.1 System Setup

Setting up the PKG consists of the following steps:

1. The PKG selects a master key, $K_{pkg}$. This key must be selected with care, as the security of the underlying system can be no more than the security inherent in this key. This key may be a symmetric key, or an asymmetric key.

2. The PKG selects an *Identity Digest Function*, $\mathsf{IDF}$. This is a pseudo-random bit function of the identity, $\mathsf{ID}$, and $K_{pkg}$ that gives an *Identity Digest Token*, $\mathsf{IDT}$ such that $\mathsf{IDT} = \mathsf{IDF}(K_{pkg}, \mathsf{ID})$.

   The $\mathsf{IDF}$ can be a symmetric-cryptographic function using the $K_{pkg}$ as some simple secret. For example, it could be a an HMAC, a CBC-MAC, or some other suitable pseudo-random bit function. The $\mathsf{IDF}$ may also be an asymmetric-cryptographic function such as RSA, in which case $K_{pkg}$ might be an appropriately strong RSA key and $\mathsf{IDT}$ is thus the result of an

RSA encryption of either ID directly or a hash of ID. Note that in this and similar cases, padding must be considered carefully to preserve the needed determinism of the IDF as it establishes a one-to-one correspondence between ID and IDT. Without a one-to-one correspondence, then this is not an IBE system at all.

It may be desirable for this selection to be part of the setup; the PKG could be built with a number of options of IDF, one selected at setup time.

Regardless of IDF selection, the resultant IDT is a limitation on the security of the IBE keys. If, for example, it were the CBC-MAC of a block cipher with a 64-bit block, then the underlying system has a birthday attack on the IDT that is probably less than the other parameters of the system. Selecting the IDF requires analysis of the overall system lest this be the security bottleneck of the system.

3. The PKG selects a deterministic pseudo-random number generator, $RNG$ that will be seeded with IDT. This $RNG$ is not the same function as IDF as it will in turn be used by a key generation function, Kgen, that generates an IBE key pair. This would be an RSA, DSA, Elgamal, or other key generation function[2]. Of course, it itself must be deterministic, as the same key must be generated any time a given identity is put into the system.

This construction has advantages beyond the simplicity of being able to use any key type within an IBE system. The security of the system relies on previously-studied components, which provides for easier security analysis. It also implicitly guards against some forms of attacks, such as collusion attacks. Breaking the $K_{pkg}$is as hard as breaking known forms of cryptography. So long as a suitable IDF function is selected, the whole Kgen process is as secure as its underlying cryptographic subsystems.

## 7.2 Key Extraction

When the PKG is requested for a key for a given ID, it follows the following process:

1. The PKG produces an IDT, such that $\mathsf{IDT} = \mathsf{IDF}(K_{pkg}, \mathsf{ID})$.
2. The PKG seeds $RNG$ with IDT.
3. The PKG generates a key with $\mathsf{Kgen}(RNG)$ to produce the appropriate IBE key pair, $IKP_{\mathsf{ID}}$.
4. If the PKG has an unauthenticated request for the given ID, then it responds with $IKP_{\mathsf{ID}_{public}}$. This happens when Bob asks for Alice's key.
5. If the PKG has an authenticated request for ID, such as when Alice asks for her own key, then the PKG responds with both $IKP_{\mathsf{ID}_{public}}$ and $IKP_{\mathsf{ID}_{private}}$.

At this point, Alice and Bob each have the appropriate piece(s) of a conventional key pair and they use it normally.

[2]Without loss of generality, the Kgen function can also be a function such as an elliptic-curve key generator. However, since one of the advantages of this design is that it produces keys that are usable within widely-used systems. When elliptic-curve systems are more widely used, it will be trivial to extend this to an IBE system based on them.

### 7.3 Encryption and Decryption

Encryption and decryption are trivial; they are simply the encryption and decryption functions of the base cryptosystem of the IBE keys. Note that if the cryptosystem is a signature-based cryptosystem such as DSA, it is signing and verification rather than encryption and decryption.

## 8 Security Limitations

As with all IBE systems, there are a number of security limitations of this system. However, in all cases the limitations of this system are no different than for other IBE systems.

### 8.1 Key Escrow Problem

IBE systems are effectively key escrow systems. It is a limitation, if not an outright flaw of IBE that the PKG holds all the parameters needed to generate any key pair, if not the key pair itself.

Consequently, Bob can never be completely assured that Alice and only Alice can decrypt a message or created a signature. In the real world this is less of a problem than it is in theory, as the security Alice's secret key is always bounded by the operational parameters of her key storage. It is undeniable, however, that an RSA key generated on a secure token is going to be more secure than one generated in a PKG.

IBE systems, including this one, may be unacceptable for some uses. If there is a legal requirement that Alice's private half of her signing key be in her possession alone, then no IBE signing system will be acceptable.

Boneh and Franklin suggest a partial solution to this problem. In their partial solution, their master key can be split using a secret-sharing system [SHAMIR79]. This has the advantage that no single entity has any of the core secret parameters. An adversary would have to compromise enough members of a set of PKGs to reconstitute the secret. Nonetheless, this is only a partial solution. At some point, the set of PKGs must reconstitute the parameters, and an adversary that sufficiently compromises the appropriate member can still get the parameters. Furthermore, since the members of the PKG set are likely to be close to identical, they are not independent in their security. If an adversary can compromise one member of the set, it is more possible if not likely that the adversary can compromise the whole set.

Another solution would be to keep the master parameters in secure hardware, or even secret-shared across a set of pieces of secure hardware. But this adds complexity on top of complexity to the system.

In this system, we accept that the IBE parts of this system are by necessity a key escrow system, but note that it can fully interoperate with another other PKI that is not a key escrow system. Furthermore, this system can be integrated with a more secure public key system to provide it with IBE features. For example, the IBE in this system gives a way that keys can be created for roles such as *Security Officer* or *Ombudsman* without pre-defining these roles or their owners prior to use. This is another advantage to merging IBE aspects into conventional PKI. Within a given

PKI, you can have parts of it that are IBE-derived, and parts that are fully-secure, edge-generated public key pairs. Moreover, they all interoperate seamlessly.

### 8.2 Security of Key Generation

The security of the keys generated by the PKG are bounded by the selection of the underlying functions as well as the scale of the PKG. If the PKG is to generate many, many keys, then factors such as the possibility of identity collision have to be taken into account as well.

This is not an intractable problem — there are many underlying functions that can be used for components of the PKG that have adequate security parameters for security. It must simply be noted that these are security design factors that are unique to an IBE system.

### 8.3 Security of Key Extraction

When Alice extracts her private key from the PKG, the PKG must deliver it to her securely. There are many ways to do this, including secure network connections such as TLS [TLS]. It also must be packaged securely (and this is another place where existing data structure systems such as certificate standards gain help). This is again, not precisely a security problem but more of where the PKG builders must take care in their delivery system.

## 9 Open Problems

IBE is not yet a mature discipline. There are a number of open problems beyond improving the security of the underlying system that are yet to be solved. Here is a short discussion of some of them.

### 9.1 Removing Key Escrow

All IBE systems, including this one, accept the fact that they are key escrow systems. However, nearly any discussion of IBE includes a class of people who consider the escrow aspect to be a severe flaw. It certainly makes the system brittle, as security of the system relies on non-mathematical security. A real-world PKG *must* be an un-hackable computer, even if that computer has advanced mathematical techniques such as secret-sharing as an additional bit of armor. It makes the simplicity that IBE gives on one axis be balanced by complexity on another.

As cryptographers and systems designers, we have accepted key escrow as a part of the playing field because if we don't, there's no IBE. In an academic paper, this is a reasonable assumption, but in the larger world, this assuming key escrow as an implicit part of the system cannot be simply brushed away.

This is a large open problem with no good solution. IBE exists for operational simplicity, but has operational complexity as a cost. Removing that cost should be the primary goal of future work, in this author's opinion.

## 9.2 Is it Possible to Eliminate Certificates?

The whole *raison d'être* for IBE is to "solve" the certificate problem. However, this means that IBE assumes that it is *possible* for a certificate to consist solely of a name and a key. In the real workd, certificates have always been more than a mere binding between a name and a key; they also carry metadata about the name, the key, parameters of use, and even metadata about the metadata.

One of the most important bits of metadata about a key is revocation data. If a name *is* a key, then it is not possible to revoke the key without revoking the name as well. The utility of Alice not having to have a certificate is small if she must revoke her email address if she loses a smart card. Furthermore, telling everyone who has Alice's email address that they must use her new one (and thus new key) is *precisely* a key revocation problem with added disadvantages for Alice.

Boneh and Franklin [BF01] suggest a clever solution to this situation. In their paper, they suggest that $\mathsf{ID}_{Alice}$ not be `"alice@hotmail.com"` but `"alice@hotmail.com || 2004"`. They even suggest that the PKG can create daily-use keys such as `"alice@hotmail.com || February 29, 2004"`.

As elegant as this solution is, it prompts other questions. Once an identity is not simply a name, but is now a name and a date, is it still Identity-Based Encryption? Phrased another way, isn't `"alice@hotmail.com || 2004"` merely a different way to code a certificate?

Implementing this solution also requires other surrounding standardization that detracts from the essential simplicity of the IBE concept. At this simplest form, you have to standardize on what a date is. This isn't difficult, but you have to do it. You must also translate malformed dates (perhaps `"2 Février 2004"` into `"02/02/2004:12:00:00.00UTC+0100"` which again detracts from the simplicity of IBE, as this is no longer something that a human being can reliably type the way that they can reliably type `"alice@hotmail.com"`. However, this is a problem that can be solve through software, an one where an on-line system has an advantage as it can canonicalize time in a central place, or even round to an internal epoch.

Previously in this paper, we discussed the algorithmic revocation problem as well. No IBE system before this one has even considered how the IBE parameters, the IBE algorithm itself, can be revoked. The fact that IBE is brittle in its reliance on central secrets makes this lack a larger open problem.

Lastly, there is no means in an identity to express variables within a cryptosystem. There can be no negotiation about acceptable block ciphers, data compression, data MACing, both start and stop date of a given key, and so on. An IBE system cannot help but be a one-size-fits-all system for these parameters. This may not be bad, it may actually be a simplifying assumption. However, expressing these concepts are part of why we have certificates despite the problems in managing them.

There are two possible approaches to dealing with this paradox — one being to make an IBE system that codes a more formal certificate and then uses that as an IBE key, such as Gentry's CBE, or this approach which adapts IBE so that it can be used within a traditional certificate system.

## 9.3 Is it Possible to Prove Ownership of a String?

When we describe how IBE works, we get to the Key Extraction phase and glibly say Alice authenticates herself to the PKG to get her private key. How?

If Alice is already an authenticated user of the PKG, this isn't very difficult. If it is to `hotmail.com` that Alice must prove ownership of `alice@hotmail.com`, this is an easy problem. If worst comes to worst, she has a password she can type in.

If it is to `brand-new-service.com` that Alice must prove ownership of `alice@hotmail.com`, it is a bit more difficult, but hardly impossible. A simple, if low-security mechanism is for `brand-new-service.com` to send her an email with some authentication token that she delivers back to `brand-new-service.com`. For those who believe this insufficiently secure, there are other protocols that are easy to devise that are more secure. For example, Alice could generate an ephemeral RSA key pair, give `brand-new-service.com` the public key, and then deliver to `brand-new-service.com` the decrypted authentication token as before. While not perfect, it's an improvement. Devising a protocol that is immune to man-in-the-middle attacks is left as an exercise to the reader.

However, if Alice must prove the ownership of `"Alice Jones"`, then we have a very difficult problem. Names are hard to express in a certificate system, and among the many criticisms of certificate systems the most basic objections concern the way they handle naming [ELLISON00]. If a name is a key, then a certificate is a key, and all the naming problems we have in certificates we have in names. Making names be keys exacerbates this problem.

If the IBE system uses other bit strings such as photographs, music, etc. as keys, proof of ownership could be arbitrarily hard, both technically and legally.

## 9.4 Performance Bottlenecks

IBE systems in general suffer from centralization on many fronts. Not only does centralization create security issues, but it also creates performance issues. The PKG may have to do large amounts of work, especially when the system uses many short-lived keys. In this system, the need for the PKG to generate keys makes more work for it. Furthermore, generating a key for some cryptosystems such as RSA require more computation than for others, such as DSA.

One possible solution to this problem is HIBE. HIBE expresses the user's identity as a composite of identities. For example, Alice's identity would be the tuple $(\mathsf{ID}_{Alice}, \mathsf{ID}_{hotmail.com})$ [GS02] [HL02]. While attractive from a performance viewpoint, it also blurs the conceptual simplicity of a name being a key. It also requires that the identities themselves have some structure in them that can form a hierarchy. HIBE also provides a partial solution to the escrow problem as no single server has the key for any hierarchical identity; an adversary must compromise more than one part of the hierarchy.

Additionally, systems that secret-split in a set of authorities could potentially also use this as a way to distribute the computational workload of IBE over the set. Nonetheless, performance is another consideration that IBE systems must take into account, and this one more than most, since there is no off-line generation of keys.

## 10 Conclusion

This presents hybrid system that combines Identity-Based features with a conventional public-key cryptosystem. Its advantages over the previous systems are that it provides interoperability with existing systems and by becoming an on-line system avoids the security problems associated with other IBE systems that permit off-line key generation. Consequently, this brings the advantages of an IBE system — that any bit string be equivalent to a public key, without the disadvantages of permitting an attacker complete knowledge of the PKG. It thus brings at this modest cost the advantages of IBE to conventional public key cryptosystems.

## References


[AY03] S. Al-Riyami and K. G. Paterson, *Certificateless Public Key Cryptography*, extended abstract in Proceedings of ASIACRYPT '03, LNCS 2894, Springer-Verlag, 2003. Full paper available in the IARC eprint archives, <http://eprint.iacr.org/2003/126/>.

[BB04] D. Boneh and X. Boyen, *Secure Identity Based Encryption Without Random Oracles*, extended abstract in Proceedings of CRYPTO '04, LNCS 3152, Springer-Verlag, 2004. Full paper available in the IACR eprint archives, <http://eprint.iacr.org/2004/173/>.

[BF01] D. Boneh and M. Franklin, *Identity-Based Encryption from the Weil Pairing*, Proceedings of CRYPTO '01, LNCS 2139, pages 213-229, Springer-Verlag, 2001.

[BIHAMCHEN04] E. Biham and R. Chen, *Near-Collisions of SHA-0*, Proceedings of CRYPTO '04, LNCS 3152, Springer-Verlag, 2004. Also available in the IACR eprint archives, <http://eprint.iacr.org/2004/146/>.

[COCKS01] C. Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, Proceedings of the 8th IMA International Conference on Cryptography and Coding, LNCS 2260, pages 360-363, Springer-Verlag, 2001.

[ELLISON00] C. Ellison, *Naming and Certificates*, Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions, ACM, <http://www.cfp2000.org/papers/ellison.pdf>.

[GENTRY03] C. Gentry, *Certificate-Based Encryption and the Certificate Revocation Problem*, Proceedings of EUROCRYPT '03, LNCS 2656, pages 272-293, Springer-Verlag 2003. Corrected version available as <http://eprint.iacr.org/2003/183/>.

[GS02] C. Gentry and A. Silverberg, *Hierarchical ID-Based Cryptography*, Proceedings of ASIACRYPT '02, LNCS 2501, pages 548-566, Springer-Verlag 2002. Also available as <http://eprint.iacr.org/2002/056/>.

[HL02] J. Horwitz and B. Lynn, *Toward Hierarchical Identity-Based Encryption*, Proceedings of EUROCRYPT '02, LNCS 2332, pages 466-481, Springer-Verlag 2002.

[JOUX04] A. Joux, *Multicollisions in Iterated Hash Functions*, Proceedings of CRYPTO '04, LNCS 3152, Springer-Verlag, 2004.

[LQ03] B. Libert and J. Quisquater, *New Identity Based Signcryption Schemes from Pairings*, IEEE Information Theory Workshop, 2003. Also available as <http://eprint.iacr.org/2003/023/>.

[LQ04] B. Libert and J. Quisquater, *What Is Possible with Identity Based Cryptography for PKIs and What Still Must Be Improved*, Proceedings of EUROPKI 2004, pages 57-70, Springer-Verlag 2004.

[MILLER56] G. A. Miller, *The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information*, The Psychological Review, 1956, vol. 63, pp. 81-97. Also available as <http://www.well.com/user/smalin/miller.html>.

[SHAMIR79] A. Shamir, *How to share a secret*, Communications of the ACM, Volume 22, Issue 11 (November 1979), pages 612-613.

[SHAMIR84] A. Shamir, *Identity-based Cryptosystems and Signature Schemes*, Proceedings of CRYPTO '84, LNCS 196, pages 47-53, Springer-Verlag, 1984.

[TLS] T. Dierks and C. Allen, *The TLS Protocol Version 1.0*, RFC2246 <http://www.ietf.org/rfc/rfc2246.txt>

[WANG04] Xiaoyun Wang and Dengguo Feng and Xuejia Lai and Hongbo Yu, *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, IACR eprint archive, <http://eprint.iacr.org/2004/199/>.

[ZHENG97] Y. Zheng, *Digital Signcryption or to achieve cost(signature & encryption) « cost(signature)+cost(encryption)*, Proceedings of CRYPTO '97, LNCS 1294, pages 165-179, Springer-Verlag, 1997.