An Introduction to Identity-based Cryptography

CSEP 590TU ▪ March 2005 ▪ Carl Youngblood

One significant impediment to the widespread adoption of public-key cryptography is its dependence on a public-key infrastructure that is shared among its users. Before secure communications can take place, both sender and receiver must generate encryption and signature keypairs, submit certificate requests along with proof of identity to a Certificate Authority (CA), and receive CA-signed certificates, which they can then use to authenticate one another and exchange encrypted messages. This process can be both time-consuming and error-prone, and is especially prohibitive for novice computer users. Frequently, individuals who can already receive encrypted email are still unable to send secure messages to others due to lack of preparedness, limited interoperability, device limitations or lack of technical competence on the receiving end. Given the need to communicate and the complexity of PKI-based cryptography, correspondence that could benefit from additional security is frequently conducted in the clear. Identity-based cryptography (IBC) seeks to reduce these barriers by requiring no preparation on the part of the message recipient. Although it provides some advantages over PKI-based approaches, it is not without its drawbacks.

History of identity-based cryptography

In 1984, Adi Shamir, of RSA notoriety, introduced the concept of identity-based cryptography [10]. Its primary innovation was its use of user identity attributes, such as email addresses or phone numbers, instead of digital certificates, for encryption and signature verification. This feature significantly reduces the complexity of a cryptography system by eliminating the need for generating and managing users' certificates. It also makes it much easier to provide cryptography to unprepared users, since messages may be encrypted for users before they interact with any system components.

At the time Shamir published his proposal he had already determined a way of using the existing RSA function for an identity-based signature (IBS) scheme, but had yet to solve the problem of identity-based encryption (IBE). This remained an open problem until 2001, when two independent lines of research (Boneh and Franklin [4], as well as Cocks [6]) arrived at solutions to the problem. Since this time, identity-based cryptography has been a heavily-researched topic in the field of cryptography [2]. In addition to academic research, commercial product offerings are also now available, most notably those of Voltage Security, Inc.

Overview of cryptographic operations

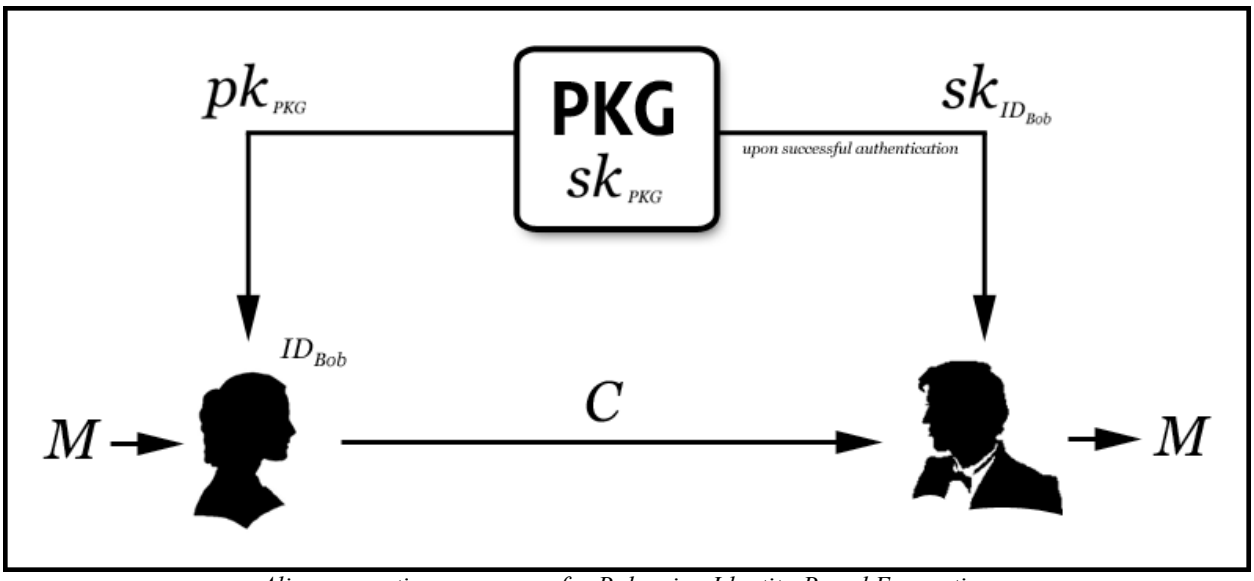
IBC relies on a trusted third party called the Private Key Generator (PKG). Before operation can begin, the PKG must generate a public/private keypair (denoted pk_{PKG} and sk_{PKG} in the following figures) and make pk_{PKG} available to users of its services. These keys are called the “master” public key and master private key, respectively.

The process of encryption and decryption proceeds as follows:

1. Alice prepares plaintext message M for Bob. She uses Bob's identity ID_{Bob} and the PKG's public key pk_{PKG} to encrypt M, obtaining ciphertext message C. Alice then sends C to Bob. Note that ID_{Bob} and pk_{PKG} were both already known to Alice before beginning the encryption

process, so she requires no prior coordination or preparation on Bob's part to encrypt a message for him.

2. Bob receives *C* from Alice. In most implementations it is assumed that *C* comes with plaintext instructions for contacting the PKG to get the private key required to decrypt it. Bob authenticates with the PKG, essentially sending it sufficient proof that ID_{Bob} belongs to him, upon which the PKG transmits Bob's private key sk_{IDBob} to him over a secure channel. If ID_{Bob} were based on an email address, for example, the PKG could send a nonce to this email address, the successful return of which might provide an acceptable level of assurance that the owner of ID_{Bob} was the one who had contacted the PKG. This nonce could be returned via an SSL hypertext link which presented Bob with a secure link for downloading his private key. For a higher level of assurance, Bob could be required to present his credentials in person and receive a compact disc containing sk_{IDBob}.
3. Bob decrypts *C* using his private key sk_{IDBob} to recover plaintext message *M*.

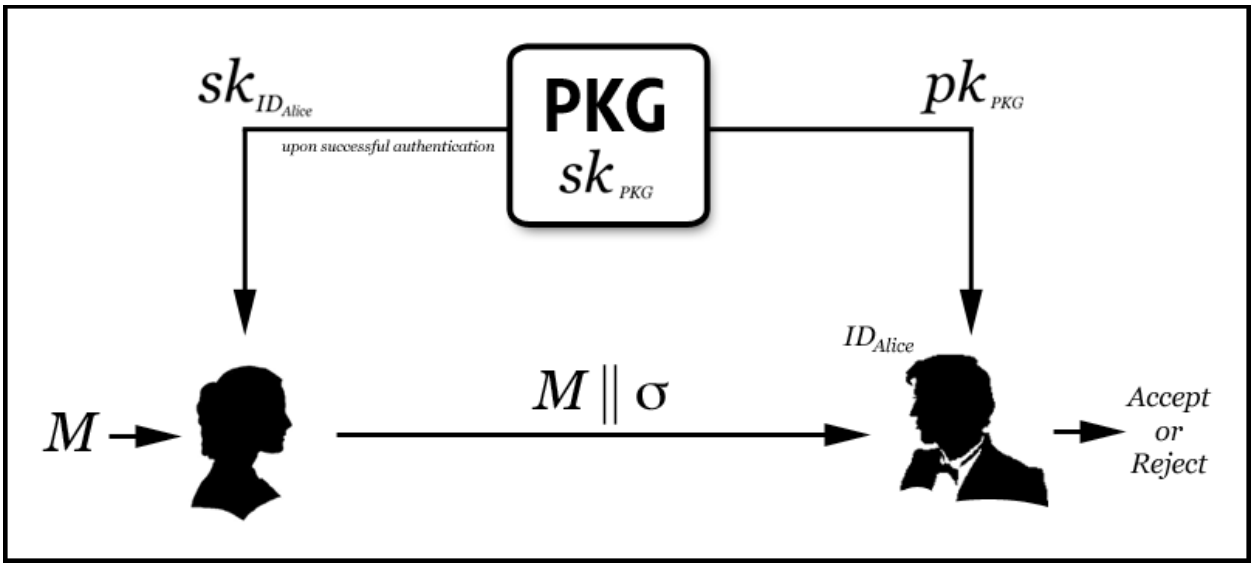


Alice encrypting a message for Bob using Identity-Based Encryption

One variation of the process described above is that the PKG can decrypt *C* for Bob and transmit it to him securely upon authentication. This is sometimes used to increase the user-friendliness of the decryption process, and it calls to attention an inherent weakness of IBC that research is currently striving to overcome: the fact that all private keys must be escrowed by the PKG. This issue will be covered in greater detail.

Identity-based signature (IBS) is essentially a mirror image of the encryption process:

1. Alice authenticates with the PKG and receives her private key $sk_{IDAlice}$.
2. Using her private key $sk_{IDAlice}$, Alice generates a signature σ for *M* and transmits it to Bob, perhaps along with encrypted message *C* above.
3. After receiving M and σ from Alice, Bob checks whether σ is a genuine signature on *M* using Alice's identity ID_{Bob} and the PKG's public key pk_{PKG}. If it is, he returns "*Accept*". Otherwise, he returns "*Reject*". Note that Bob doesn't need to have any type certificate for Alice.



Alice signing a message for Bob using Identity-Based Signature

Security of identity-based cryptography

The vast majority of proposed identity-based cryptography schemes, and certainly all of those discovered so far that are computationally efficient, are based on mathematical functions called *bilinear nondegenerate maps*. A bilinear nondegenerate map is a function pairing elements from one cyclic group to another of the same prime order, where the discrete log problem is hard in the first group [3].

The security of identity-based cryptography is based on the assumption that the particular bilinear maps chosen are one-way functions, meaning it is easy to calculate their result given a pair of operands but hard to calculate the inverse. This property is often referred to as the *Bilinear Diffie-Hellman Assumption*, since the Bilinear Diffie-Hellman problem is reducible (algorithmically equivalent) to the discrete-log or inverse operation for these bilinear maps [11].

In simplified notation, a bilinear map is a pairing that has the property:

$\text{Pair}(a \cdot X, b \cdot Y) = \text{Pair}(b \cdot X, a \cdot Y)$

In two of the more well-known IDE systems, the Weil (pronounced *vay*, rhyming with the English word *way*) and Tate pairings, the · operator above refers to multiplication of a point on an elliptic curve by integers [9]. Although the multiplication operation, such as calculating $a \cdot X$, is easy, finding a given X and $a \cdot X$ is computationally infeasible.

Boneh and Franklin were the first to propose a viable IDE system based on the Weil pairing in 2001, nearly two decades after Shamir's original proposal. Since that time a number of other pair-based IDE and IDS systems have been proposed. Since most of these are pairing-based, identity-based cryptography is often called *pairing-based cryptography*.

Cryptographic operations in the Boneh and Franklin IDE system are conducted as follows. Note

that some details of the math involved in elliptic curves have been omitted for clarity's sake [9]:

Setup: The PKG picks an elliptic curve, a secret s and a point P on the curve using a random number generator. It then publishes P and $s \cdot P$ as the master public key.

Encryption: Alice hashes the chosen identity attribute for Bob to a point ID_{Bob} on the elliptic curve. She then picks a random r and calculates a key k:

$k = \text{Pair}(r \cdot ID_{Bob}, s \cdot P)$

Alice then sends $E_k[M]$ and $r \cdot P$ to Bob.

Decryption: Bob may not yet have a private key. To get it, he authenticates with the PKG, which calculates $s \cdot ID_{Bob}$ and returns it to him over a secure channel. This is his private key. After receiving $E_k[M]$ and $r \cdot P$ from Alice, Bob can recover the key k by calculating:

$k = \text{Pair}(s \cdot ID_{Bob}, r \cdot P)$

This is possible because of the properties of bilinear maps. Bob can then use k to decrypt the message. No one else (besides the PKG) can calculate k because only Bob knows $s \cdot ID_{Bob}$.

Even though Shamir had already provided one possible identity-based signature system based on RSA in his seminal proposal, other researchers have since discovered pairing-based IBS systems to complement the pairing-based encryption systems. One of the first such systems was proposed by Boneh, Lynn and Shacham [5].

Pros and Cons of identity-based cryptography

Some of the advantages of IBC have already been explained, but here is a summary:

1. No preparation is required on the part of the recipient to receive an encrypted message. This is arguably the most compelling feature of IDC.
2. No need to managing a public key infrastructure, including CRL management.
3. IBC's inherent key escrow feature means decryption and signature can take place on the server. While this is a disadvantage (especially in IDS because it eliminates non-repudiation in most cases), it also makes certain other features possible that are not possible in PKI-based systems where the signer is in possession of his/her private key, such as:
   i. “Chameleon” signatures, in which only the designated recipient is capable of asserting a signature's validity [1].
   ii. Improved user-friendliness by having the PKG handle cryptographic operations for the user and requiring no client-side installation. This can be especially powerful in the case where an enterprise wants to adopt a policy whereby all messages of a certain sensitivity level are automatically encrypted and/or signed. An administrator can specify the policies that govern whether a message will be signed or encrypted using tools like a keyword search of the message content, a time range, or a regular expression match on the sender or recipient, and email users do not need to modify their behavior. [7]
   iii. If users do not have to receive their private key, it can be kept on the PKG, which often has a much higher level of security than a user's workstation.

4. No PKI means less public information about your enterprise need be revealed to those who do not have a need to know. Each application or person connecting to an enterprise's certificate database could theoretically discover a great deal of information about a company's infrastructure or hierarchy. For large companies where some employees work on sensitive projects or where many employees only interact with their close colleagues on a daily basis, not needing to access a certificate database could be beneficial. [7]

The most notable disadvantage of IBC is its inherent key escrow property. While it has already been noted that this can be an advantage in some cases, most IDC adopters would like to be able to decide whether or not they want this feature. It should be noted that many organizations already employ encryption key escrow, to be able to recover a user's encrypted data in the event his or her private key is lost. This should be taken into account when analyzing the security of IBC systems. The practical difference, therefore, between IBC and most PKI systems is that PKI systems do not escrow users' signature keys. This allows for better non-repudiation, which is an essential feature of digital signature schemes. A number of IBC variants are also being developed that eliminate or mitigate the key escrow feature, including certificate-based encryption, secure key issuing cryptography and certificateless cryptography. [8] In secure key issuing, for example, the PKG's level of trust is reduced by spreading the master keys across multiple PKGs. While this increases the system's security, it also decreases performance. [2]

In our discussion of non-repudiation, it should also be noted that even PKI systems don't provide for a perfect level of non-repudiation, since there is always a time frame before a compromised key is reported to the CA. In IBC systems there is also still some level of non-repudiation, but it is tied to the level of trust that the PKG is not signing messages or is only signing messages at the user's request.

IBC in its most basic form also lacks key revocation. Suppose, for example, that Bob's private key is compromised. The key was associated with his public email address, which has been serving as his public key. Does Bob need to change email addresses now that his private key has been compromised? Even worse, suppose Bob's private key is associated with some type of biometric data. Does Bob forever lose the ability to use his thumbprint, for example, as his identity, because his private key was compromised? A simple solution to this problem was proposed by Boneh and Franklin [4], who suggested that the ID component could be concatenated with a validity timestamp. This would mean that the public key would only be valid until the timestamp expired, which would place a limit on the amount of damage that could be done by a key security breach.

One other important drawback of IBC systems is the high level of assurance required in the PKG. Since the PKG holds all private keys, it requires a higher level of assurance and availability than a CA. CAs may be kept disconnected from a network, but the PKG must be available to send users their private keys, further increasing its vulnerability to attack. For this reason, extra care must be taken to secure PKGs above and beyond the high level of security already required for CAs.

Implementations of identity-based cryptography

Boneh and Franklin, along with other researchers, developed a C++/based IBE implementation

published under an MIT-style license, called the “Stanford IBE System” [2].

Shamus Software also developed another C++-based cryptographic library called “MIRACL” which follows Boneh and Franklin's IBE scheme.

The most notable commercial implementation of IBE is published by Voltage Security, Inc. [8] They offer plug-ins for a number of popular mail clients, including Microsoft Outlook. Proofpoint, Inc. has licensed Voltage's software to provide value-addons, such as policy-based automatic outbound email encryption. [7]

Summary

The lack of widespread adoption of conventional PKI systems is a persuasive argument for their excessive complexity. Identity-based systems offer some significant advantages over PKI, especially in their increased user-friendliness, though they do not come without some drawbacks. For its advocates, IBC provides a better compromise between security and complexity than previous systems. In spite of its weaknesses, the high level of research being conducted in this field is a testimony of its potential to overcome some of the current problems plaguing cryptographers.

References

[1] G. Ateniese and B. Medeiros, *Identity-based Chameleon Hash and Applications*, Financial Cryptography – Proceedings of FC 2004, LNCS, Springer-Verlag. http://www.cs.jhu.edu/~ateniese/papers/id-chameleon.pdf.

[2] J. Baek, J. Newmarch, R. Sfavi-Naini, W. Susilo, *A Survey of Identity-Based Cryptography*, http://jan.netcomp.monash.edu.au/publications/auug_id_survey.pdf.

[3] *Bilinear Pairings.* http://rooster.stanford.edu/~ben/maths/ep/pairing.php

[4] D. Boneh and M. Franklin, *Identity-Based Encryption from the Weil Pairing*, Proceedings of CRYPTO 2001, LNCS 2139, pages 213–229, Springer-Verlag, 2001. http://crypto.stanford.edu/~dabo/papers/ibe.pdf.

[5] D. Boneh, B. Lynn, H. Shacham, *Short Signatures from the Weil Pairing*, Asiacrypt, Lecture Notes in Computer Science, vol. 2248, pages 514+, 2001. http://citeseer.ist.psu.edu/boneh01short.html.

[6] C. Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, Cryptography and Coding - Institute of Mathematics and Its Applications International Conference on Cryptography and Coding – Proceedings of IMA 2001, LNCS 2260, pages 360–363, Springer-Verlag, 2001. http://www.cesg.gov.uk/site/ast/idpkc/media/ciren.pdf.

[7] *Encryption Made Easy: The Advantages of Identity-Based Encryption*, Proofpoint Inc. http://www.proofpoint.com/downloads/WP-Proofpoint-Encryption-Made-Easy.pdf.

[8] *Identity-based encryption*. http://en.wikipedia.org/wiki/Identity_based_encryption.

[9] *Identity-based encryption.* http://www.voltage.com/technology/ibe.htm

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

[11] Yacobi, Yacov, *A Note on the Bi-Linear Diffie-Hellman Assumption,* Cryptology ePrint Archive, Report 2002/113, 2002. http://citeseer.ist.psu.edu/yacobi02note.html.