Identity-Based Encryption: A Closer Look

By Luther Martin
martin@voltage.com

A new public-key encryption technology called identity-based encryption (IBE) allows you to calculate a public key directly from a user's identity. By calculating public keys instead of generating them randomly, many of the difficulties that make encryption technology difficult to deploy and maintain are eliminated, making encrypted communications much easier to implement than in the past.

Why Encrypt Your Data?

The recent changes in the regulatory environment have made businesses more interested in secure communications, whether they are over e-mail, instant messaging, or other means. The Health Insurance Portability and Accountability Act (HIPAA) requires the protection of any individually identifiable health care information, the Gramm-Leach-Bliley Act (GLBA) includes requirements to protect consumers' personal financial information, and the Sarbanes-Oxley Act (SOX) requires many companies to make statements about their IT security practices and regulatory compliance in their annual reports. In each of these cases, there is no explicit requirement for encrypting data, but encryption is certainly a quick and easy way to meet the regulatory requirements in a way that your auditors will accept. And California Senate Bill 1386 (SB 1386), the Security Breach Information Act, requires businesses to notify California residents of any unauthorized disclosure of unencrypted personal information, making encryption an easy way to reduce the cost of responding to security breaches.

Although the use of encryption has become widely adopted in a few applications, such as SSL and VPNs, businesses have been reluctant to deploy technologies that provide end-to-end encryption of communications, so there is still a great deal of sensitive information being transmitted in the clear over the Internet. And while the idea of encryption is appealing to many businesses, the practical difficulties of deploying and supporting it have kept its acceptance out of the mainstream.

Practical Difficulties with Encryption

Symmetric encryption algorithms like Triple DES and AES use the same cryptographic key to encrypt and decrypt, and when used correctly, can provide virtually unbreakable security for the information that they are used to protect. But their use creates another problem, since it is difficult to securely distribute a symmetric key to both the sender and the recipient of an encrypted message. The difficulties that come with distributing cryptographic keys do not stop the technology from being widely used, however, and today you will find symmetric encryption widely used in both ATM machines and point-of-sale (POS) devices.

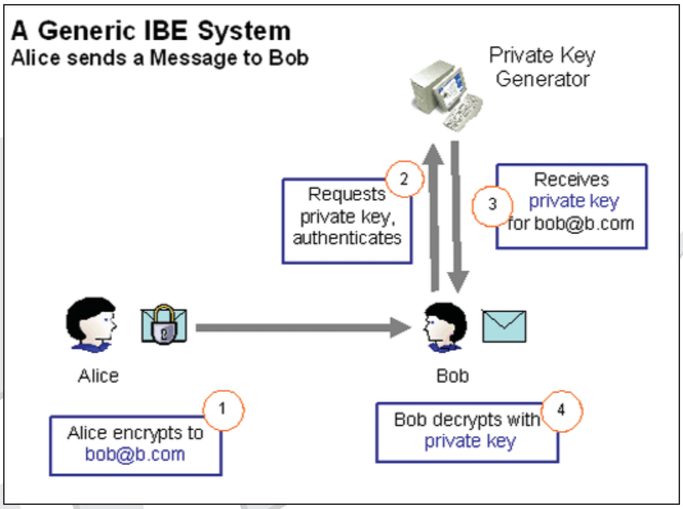
A Generic IBE System
Alice sends a Message to Bob
Private Key Generator
2
Requests private key, authenticates
3
Receives private key for bob@b.com
Alice
Bob
1
Alice encrypts to bob@b.com
4
Bob decrypts with private key

Figure 1: Operation of a Generic IBE System

The common element that lets ATMs and POS devices use encryption so easily is that these devices only communicate with a single well-known system. Unfortunately, very few business applications are this simple. In more complex systems, each user needs a different key to communicate securely with each other user; otherwise users would be able to decrypt messages that were not intended for them. And since cryptographic keys need to be changed frequently, just like passwords do, having a large number of users will also give us a large number of keys that we may need to securely update on a regular basis.

Public-key cryptography was invented to solve the symmetric key distribution problem. In a public-key system, each person gets a pair of keys, one called the *public key* and the other called the *private key*, and the two keys are mathematically related so that information that is encrypted with the public key can only be decrypted with the corresponding private key and *vice versa*. So if the public key is made available to everyone while the private key is kept secret, anyone can encrypt a message to someone by using their public key, but only the intended recipient can decrypt the message by using the corresponding private key. So by encrypting a symmetric key to a person using their public key, we can solve the symmetric key distribution problem, but we also acquire a number of additional difficulties that using public-key cryptography brings.

Public-key cryptography has its own key distribution problem. The original concept of using public keys[1] assumed that they would be easily accessible in a public directory of some sort, much like a universal phone book, but a practical implementation of this idea has evaded technologists for close to 30 years. Public keys are typically distributed in digital certificates which are stored in an LDAP directory, and the practical difficulties of using LDAP in this way are well-documented[2,3] and relate to the complexity and difficulty of using the protocol to support this particular application. There is no fundamental problem with public-key cryptography in this case, but the protocol that technology vendors have adopted to support its implementation has proven to provide obstacles to this particular use.

Key recovery is another problem that makes deployment of public-key technology difficult. Recovery of a private key is often required, for example, when the original recipient of an encrypted message is not available but you still need to decrypt the message. The recipient may no longer work for your company, or may just be away on vacation. Regulatory concerns may require you to archive an unencrypted copy of all messages that are sent by your organization at the same time that end-to-end encryption is required. Or you may want to scan all messages for malicious content like viruses, spam or phishing attacks. In any of these cases you would need access to the same private key that the recipient of the message has, but since private keys are randomly generated, the only way to achieve this is to securely archive copies of all of the private keys. This leads to the significant cost and complexity of ensuring that backups are complete and disaster-recovery plans are comprehensive in addition to the cost of the infrastructure needed to achieve this.

Identity-Based Encryption

The idea of IBE was first introduced by Adi Shamir in 1984[4] as a way to eliminate the difficulties associated with using a public-key system. Shamir's idea was to create a way of calculating a public key from a user's identity, and he described an identity-based system in the following way:

> *"An identity-based scheme resembles an ideal mail system: If you know somebody's name and address you can send him messages that only he can read, and you can verify the signatures that only he could have produced. It makes the cryptographic aspects of the communication almost transparent to the user, and it can be used effectively even by laymen who know nothing about keys or protocols."*

In practice, the form of the identity that is used to calculate an IBE key depends on the application. For encrypting e-mail, a string that represents the e-mail address of the recipient is a good choice, but in other applications, a phone number, a device serial number, an IP address or a MAC address might be more logical; any identity that is globally unique can be used.

Some 17 years after Shamir's original statement of the problem, two IBE schemes were invented that proved to be both secure and practical: one was invented by Clifford Cocks[5], the other by Dan Boneh and Matt Franklin[1]. The scheme invented by Cocks relies on the *quadratic residuosity* problem for its security, a mathematical problem that relates to determining whether or not a particular integer has a square root module or a large composite integer. The scheme invented by Boneh and Franklin relies on the *bilinear Diffie-Hellman problem* for its security and uses a complicated mathematical transformation called the *Tate pairing*. With each of these IBE schemes, the general architecture of a complete system is the same and is shown in Figure 1. In each case, the sender calculates the public key of the recipient and the recipient then has to authenticate to a private key generator (PKG) to obtain the corresponding private key.

The Cocks IBE scheme has one limitation to its practicality, in that it requires the transmission of two large integers to send each bit of a secure message. So to get the same level security that using a 1,024-bit RSA modulus provides, we need to send two different 1,024-bit numbers, increasing the bandwidth required by a factor of 2,048. Thus an 80-bit symmetric key will require the transmission of 20KB (80 bits x 2048 = 163,840 bits or 20K bytes) of data to securely transmit an 80-bit key. This much overhead may seem impractical for encrypting routine business communications, but it is probably less of a concern if we use the scheme for exchanging keying material that only needs to be done infrequently. This feature of the Cocks IBE scheme has prevented it from gaining widespread acceptance, and current IBE implementations and research now focus on pairing-based schemes, like Boneh-Franklin.

In Figure 1 we see the familiar two users that we use to describe the operation of cryptographic systems: Alice and Bob. Using an IBE system, for Alice to send an encrypted message to Bob, she first calculates Bob's public key and then uses this public key to encrypt the message. When Bob receives the encrypted message, he then authenticates himself to a PKG, which then securely sends the corresponding private key to him. Once he has the private key, he can then decrypt the message and view its contents.

Note that the public key that Alice calculates does not need to be derived from just Bob's e-mail address. Instead, Alice could have just as easily calculated a private key for "bob@b.com|role=doctor," and if we require the authentication step that Bob performs with the PKG to include proving that he is indeed a doctor, we can use this technique to easily enforce role-based access to encrypted information.

Advantages of Using IBE

IBE is a public-key technology, so it has all the benefits that other public-key technologies have, but it also brings other benefits, since IBE keys are calculated instead of being randomly generated. Since we can calculate a key for any recipient, there is no pre-enrollment required for users of an IBE system. Since we calculate keys, there is no requirement for looking up public keys, and one of the big practical difficulties that has been associated with public-key cryptography is no longer an issue. And since we calculate a user's private key when he initially requests it, we can easily recalculate it at other times, giving us built-in key recovery capability, an essential capability for an encryption system to have for it to be used by businesses.

A useful side-effect of built-in key recovery is that it is easy to integrate IBE encryption with message hygiene technologies, making it feasible to actually scan encrypted messages for malicious content like viruses, spam or phishing attacks. To implement this we just need to give a mail gateway permission to recover private keys from a PKG. Then the gateway can decrypt any encrypted messages, perform the content filtering that its security policy requires, and then re-encrypt the messages and forward them to their destination.

Being able to calculate public keys is particularly useful when you need to communicate securely but you do not know beforehand with whom you will need to communicate. For example, many different government agencies may need to respond to a disaster, but the participants in a disaster response depend on both the nature and location of the disaster. Some responses might require the Department of Energy while others will not. State and local law enforcement are involved in many disaster responses, but since we do not know beforehand where a disaster is going to take place, we do not know which law enforcement agencies will need to respond, and it is certainly not feasible to issue digital certificates to every person who might be involved in a disaster response of some kind.

Using IBE, it is easy to communicate with a person who has not already enrolled in our system. All we need to do is calculate the public key for the recipient and then use that key to encrypt a message to them. Then once the recipient of the encrypted message authenticates himself to the PKG and gets his private key, we have created a secure communication channel. Techniques like e-mail answerback can be used in situations like this to authenticate users, allowing us to easily create *ad hoc* groups that can communicate securely, but with minimal set-up cost, making it ideal for use in situations where low-overhead dynamic groups need to be created.

Further Reading

- ▲ A case study involving encryption in which IBE was determined to be a good solution is available at: http://www.hpl.hp.com/techreports/2003/HPL-2003-203.pdf (an interesting case study that describes some of the obstacles to using encryption to solve a real business problem)
- ▲ A presentation at the 1st Annual PKI Research Workshop that describes the potential uses of IBE is available at: http://www.cs.dartmouth.edu/~pki02/Levy/slides.ppt (a good overview of the potential uses of IBE in many settings)
- ▲ The results of using IBE for encrypted communications in a military homeland security exercise is available at: https://www.cwid.js.mil/public/cwid05fr/htmlfiles/u109intr.html (describes the use of IBE-based encrypted mail to and from Outlook and BlackBerry handheld devices)
- ▲ The mathematics behind IBE is an active area of research by cryptographers. The entire Summer 2004 issue (Vol. 17, No. 3) of the *Journal of Cryptology* was devoted to the subject (this is hard-core mathematics, and is probably not suitable for a general audience)
- ▲ The Pairing-Based Crypto Lounge, a Web site devoted to pairing-based cryptography and its applications is available at: http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html (this Web site lists over 200 publications, and is probably the best on-line source of information on IBE and related technologies)
- ▲ A news story about using IBE on smart cards is available at: http://www.pcw.co.uk/news/1159232 (just a news article without much technical detail)
- ▲ A Web page of links to sites with information on IBE is available at: http://ntrg.cs.tcd.ie/~argp/ibe.html (a good collection of useful links)

Summary

IBE easily solves some of the problems that have traditionally made implementing and supporting encryption technology difficult and expensive. It is a public-key technology, so it addresses the difficulties associated with using symmetric key technology, but it also addresses many of the difficulties that traditional public key systems have. By calculating public and private keys directly from an identity, IBE avoids many of the troubles associated with using digital certificates. It also has useful capabilities like easy integration with mail hygiene technologies, and its implementation can provide an easy solution to the challenges that regulations like HIPAA, GLBA, SOX and SB 1386 have introduced into the corporate IT security world.

Luther Martin is a Principal Engineer with Voltage Security. His 18-year career in cryptography has included government agencies, the Big 4 consulting world, and security product companies.

[1] W. Diffie and M.E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, Vol. 22, No. 6, pp. 644-654, 1976.

[2] D. Chadwick. "Deficiencies in LDAP when used to support a Public Key Infrastructure," Communications of the ACM, Vol. 46, No. 3, pp. 99-104, 2003.

[3] P. Gutmann. "How to build a PKI that works," 3rd Annual PKI R&D Workshop Proceedings, NISTIR 7122, 2004.

[4] A. Shamir. "Identity-based cryptosystems and signature schemes," Proceedings of CRYPTO 84 on Advances in Cryptology, pp. 47-53, 1985.

[5] C. Cocks. "An identity based encryption scheme based on quadratic residues," Proceedings of the Eighth IMA International Conference on Cryptography and Coding, pp. 360-363, 2001.

[6] D. Boneh, M. Franklin. "Identity based encryption from the Weil pairing," SIAM Journal of Computing, Vol. 32, No. 3, pp. 586-615, 2003.