An Introduction to Cryptography

Network Associates, Inc. (408) 988-3832 main
3965 Freedom Circle
Santa Clara, CA 95054
http://www.nai.com

info@nai.com

LIMITED WARRANTY

Limited Warranty. Network Associates warrants that for sixty (60) days from the date of original purchase the media (for example diskettes) on which the Software is contained will be free from defects in materials and workmanship.

Customer Remedies. Network Associates' and its suppliers' entire liability and your exclusive remedy shall be, at Network Associates' option, either (i) return of the purchase price paid for the license, if any, or (ii) replacement of the defective media in which the Software is contained with a copy on nondefective media. You must return the defective media to Network Associates at your expense with a copy of your receipt. This limited warranty is void if the defect has resulted from accident, abuse, or misapplication. Any replacement media will be warranted for the remainder of the original warranty period. Outside the United States, this remedy is not available to the extent Network Associates is subject to restrictions under United States export control laws and regulations.

Warranty Disclaimer. To the maximum extent permitted by applicable law, and except for the limited warranty set forth herein, THE SOFTWARE IS PROVIDED ON AN “AS IS” BASIS WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. WITHOUT LIMITING THE FOREGOING PROVISIONS, YOU ASSUME RESPONSIBILITY FOR SELECTING THE SOFTWARE TO ACHIEVE YOUR INTENDED RESULTS, AND FOR THE INSTALLATION OF, USE OF, AND RESULTS OBTAINED FROM THE SOFTWARE. WITHOUT LIMITING THE FOREGOING PROVISIONS, NETWORK ASSOCIATES MAKES NO WARRANTY THAT THE SOFTWARE WILL BE ERROR-FREE OR FREE FROM INTERRUPTIONS OR OTHER FAILURES OR THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NETWORK ASSOCIATES DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT WITH RESPECT TO THE SOFTWARE AND THE ACCOMPANYING DOCUMENTATION. SOME STATES AND JURISDICTIONS DO NOT ALLOW LIMITATIONS ON IMPLIED WARRANTIES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. The foregoing provisions shall be enforceable to the maximum extent permitted by applicable law.

Table of Contents

Preface

Cryptography is the stuff of spy novels and action comics. Kids once saved up Ovaltine™ labels and sent away for Captain Midnight's Secret Decoder Ring. Almost everyone has seen a television show or movie involving a nondescript suit-clad gentleman with a briefcase handcuffed to his wrist. The word "espionage" conjures images of James Bond, car chases, and flying bullets.

And here you are, sitting in your office, faced with the rather mundane task of sending a sales report to a coworker in such a way that no one else can read it. You just want to be sure that your colleague was the actual and only recipient of the email and you want him or her to know that you were unmistakably the sender. It's not national security at stake, but if your company's competitor got a hold of it, it could cost you. How can you accomplish this?

You can use cryptography. You may find it lacks some of the drama of code phrases whispered in dark alleys, but the result is the same: information revealed only to those for whom it was intended.

Who should read this guide

This guide is useful to anyone who is interested in knowing the basics of cryptography, and explains the terminology and technology you will encounter as you use PGP products. You will find it useful to read before you begin working with cryptography.

How to use this guide

This guide describes how to use PGP to securely manage your organization's messages and data storage.

Chapter 1, "The Basics of Cryptography," provides an overview of the terminology and concepts you will encounter as you use PGP products.

Chapter 2, "Phil Zimmermann on PGP," written by PGP's creator, contains discussions of security, privacy, and the vulnerabilities inherent in any security system, even PGP.

For more information

For information on technical support and answers to other product related questions you might have, please see the What's New file accompanying this product.

Related reading

Here are some documents that you may find helpful in understanding cryptography:

Non-Technical and beginning technical books

- "*Cryptography for the Internet*," by Philip R. Zimmermann. Scientific American, October 1998. This article, written by PGP's creator, is a tutorial on various cryptographic protocols and algorithms, many of which happen to be used by PGP.
- "*Privacy on the Line*," by Whitfield Diffie and Susan Eva Landau. MIT Press; ISBN: 0262041677. This book is a discussion of the history and policy surrounding cryptography and communications security. It is an excellent read, even for beginners and non-technical people, and contains information that even a lot of experts don't know.
- "*The Codebreakers*," by David Kahn. Scribner; ISBN: 0684831309. This book is a history of codes and code breakers from the time of the Egyptians to the end of WWII. Kahn first wrote it in the sixties, and published a revised edition in 1996. This book won't teach you anything about how cryptography is accomplished, but it has been the inspiration of the whole modern generation of cryptographers.
- "*Network Security: Private Communication in a Public World*," by Charlie Kaufman, Radia Perlman, and Mike Spencer. Prentice Hall; ISBN: 0-13-061466-1. This is a good description of network security systems and protocols, including descriptions of what works, what doesn't work, and why. Published in 1995, it doesn't have many of the latest technological advances, but is still a good book. It also contains one of the most clear descriptions of how DES works of any book written.

Intermediate books

- "*Applied Cryptography: Protocols, Algorithms, and Source Code in C*," by Bruce Schneier, John Wiley & Sons; ISBN: 0-471-12845-7. This is a good beginning technical book on how a lot of cryptography works. If you want to become an expert, this is the place to start.

- "*Handbook of Applied Cryptography,*" by Alfred J. Menezes, Paul C. van Oorschot, and Scott Vanstone. CRC Press; ISBN: 0-8493-8523-7. This is the technical book you should read after Schneier's book. There is a lot of heavy-duty math in this book, but it is nonetheless usable for those who do not understand the math.
- "*Internet Cryptography,*" by Richard E. Smith. Addison-Wesley Pub Co; ISBN: 0201924803. This book describes how many Internet security protocols work. Most importantly, it describes how systems that are designed well nonetheless end up with flaws through careless operation. This book is light on math, and heavy on practical information.
- "*Firewalls and Internet Security: Repelling the Wily Hacker,*" by William R. Cheswick and Steven M. Bellovin. Addison-Wesley Pub Co; ISBN: 0201633574. This book is written by two senior researchers at AT&T Bell Labs and is about their experiences maintaining and redesigning AT&T's Internet connection. Very readable.

Advanced books

- "*A Course in Number Theory and Cryptography,*" by Neal Koblitz. Springer-Verlag; ISBN: 0-387-94293-9. An excellent graduate-level mathematics textbook on number theory and cryptography.
- "*Differential Cryptanalysis of the Data Encryption Standard,*" by Eli Biham and Adi Shamir. Springer-Verlag; ISBN: 0-387-97930-1. This book describes the technique of differential cryptanalysis as applied to DES. It is an excellent book for learning about this technique.

The Basics of Cryptography 1

When Julius Caesar sent messages to his generals, he didn't trust his messengers. So he replaced every A in his messages with a D, every B with an E, and so on through the alphabet. Only someone who knew the “shift by 3” rule could decipher his messages.

And so we begin.

Encryption and decryption

Data that can be read and understood without any special measures is called *plaintext* or *cleartext*. The method of disguising plaintext in such a way as to hide its substance is called *encryption*. Encrypting plaintext results in unreadable gibberish called *ciphertext*. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting ciphertext to its original plaintext is called *decryption*.

Figure 1-1 illustrates this process.

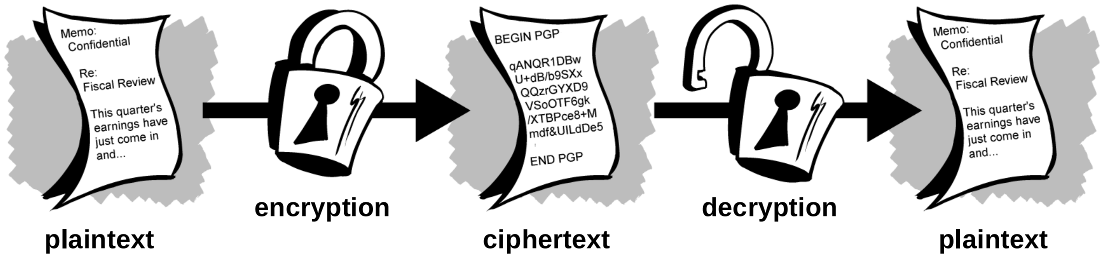

Figure 1-1. Encryption and decryption

What is cryptography?

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient.

While cryptography is the science of securing data, *cryptanalysis* is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called *attackers*.

Cryptology embraces both cryptography and cryptanalysis.

Strong cryptography

“There are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files. This book is about the latter.”

--Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C.

PGP is also about the latter sort of cryptography.

Cryptography can be *strong* or *weak*, as explained above. Cryptographic strength is measured in the time and resources it would require to recover the plaintext. The result of *strong cryptography* is ciphertext that is very difficult to decipher without possession of the appropriate decoding tool. How difficult? Given all of today’s computing power and available time—even a billion computers doing a billion checks a second—it is not possible to decipher the result of strong cryptography before the end of the universe.

One would think, then, that strong cryptography would hold up rather well against even an extremely determined cryptanalyst. Who’s really to say? No one has proven that the strongest encryption obtainable today will hold up under tomorrow’s computing power. However, the strong cryptography employed by PGP is the best available today. Vigilance and conservatism will protect you better, however, than claims of impenetrability.

How does cryptography work?

A *cryptographic algorithm*, or *cipher*, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a *key*—a word, number, or phrase—to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

A cryptographic algorithm, plus all possible keys and all the protocols that make it work comprise a *cryptosystem*. PGP is a cryptosystem.

Conventional cryptography

In conventional cryptography, also called *secret-key* or *symmetric-key* encryption, one key is used both for encryption and decryption. The Data Encryption Standard (DES) is an example of a conventional cryptosystem that is widely employed by the Federal Government. *Figure 1-2* is an illustration of the conventional encryption process.

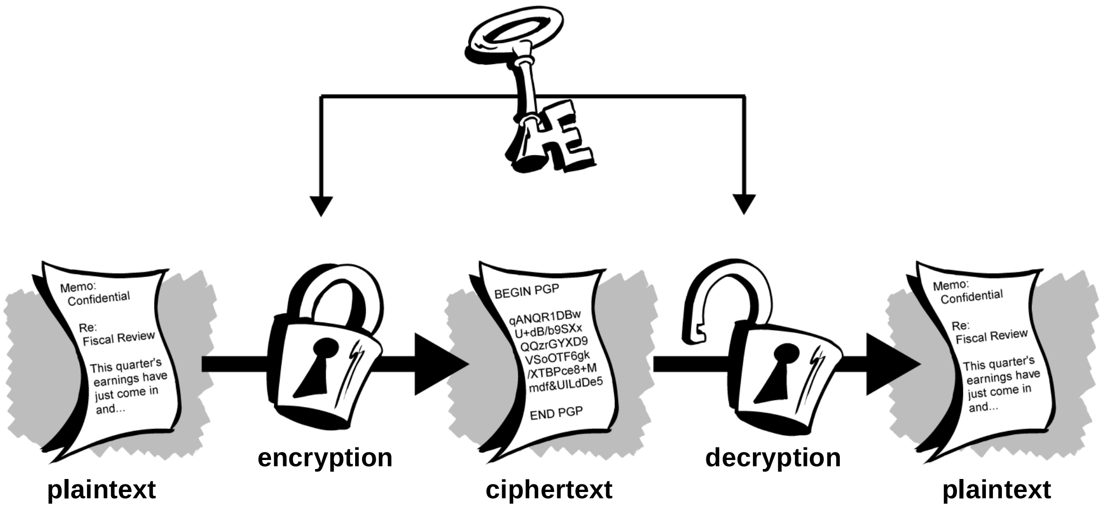

Figure 1-2. Conventional encryption

Caesar's Cipher

An extremely simple example of conventional cryptography is a substitution cipher. A substitution cipher substitutes one piece of information for another. This is most frequently done by offsetting letters of the alphabet. Two examples are Captain Midnight's Secret Decoder Ring, which you may have owned when you were a kid, and Julius Caesar's cipher. In both cases, the algorithm is to offset the alphabet and the key is the number of characters to offset it.

For example, if we encode the word "SECRET" using Caesar's key value of 3, we offset the alphabet so that the 3rd letter down (D) begins the alphabet.

So starting with

ABCDEFGHIJKLMNOPQRSTUVWXYZ

and sliding everything up by 3, you get

DEFGHIJKLMNOPQRSTUVWXYZABC

where D=A, E=B, F=C, and so on.

Using this scheme, the plaintext, “SECRET” encrypts as “VHFUHW.” To allow someone else to read the ciphertext, you tell them that the key is 3.

Obviously, this is exceedingly weak cryptography by today’s standards, but hey, it worked for Caesar, and it illustrates how conventional cryptography works.

Key management and conventional encryption

Conventional encryption has benefits. It is very fast. It is especially useful for encrypting data that is not *going* anywhere. However, conventional encryption alone as a means for transmitting secure data can be quite expensive simply due to the difficulty of secure key distribution.

Recall a character from your favorite spy movie: the person with a locked briefcase handcuffed to his or her wrist. What is in the briefcase, anyway? It’s probably not the missile launch code / biotoxin formula / invasion plan itself. It’s the *key* that will decrypt the secret data.

For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier, the Bat Phone, or some other secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key. From DES to Captain Midnight’s Secret Decoder Ring, the persistent problem with conventional encryption is *key distribution*: how do you get the key to the recipient without someone intercepting it?

Public key cryptography

The problems of key distribution are solved by *public key cryptography*, the concept of which was introduced by Whitfield Diffie and Martin Hellman in 1975. (There is now evidence that the British Secret Service invented it a few years before Diffie and Hellman, but kept it a military secret—and did nothing with it.)[1]

Public key cryptography is an asymmetric scheme that uses a *pair* of keys for encryption: a *public key*, which encrypts data, and a corresponding *private*, or *secret key* for decryption. You publish your public key to the world while keeping your private key secret. Anyone with a copy of your public key can then encrypt information that only you can read. Even people you have never met.

1. J H Ellis, The Possibility of Secure Non-Secret Digital Encryption, CESG Report, January 1970. [CESG is the UK’s National Authority for the official use of cryptography.]

It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information.

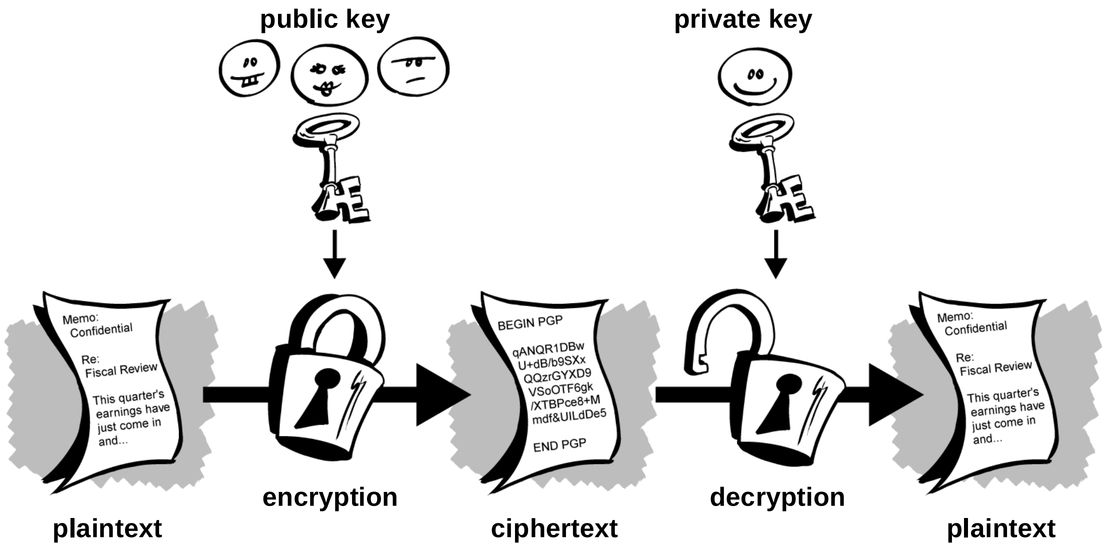
public key
private key
Memo: Confidential Re: Fiscal Review This quarter's earnings have just come in and...
encryption
BEGIN PGP qANQR1DBw U+dB/b9SXx QQzrGYXD9 VSoOTF6gk /XTBPce8+M mdf&UILdDe5 END PGP
decryption
Memo: Confidential Re: Fiscal Review This quarter's earnings have just come in and...
plaintext
ciphertext
plaintext

Figure 1-3. Public key encryption

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Some examples of public-key cryptosystems are Elgamal (named for its inventor, Taher Elgamal), RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman), Diffie-Hellman (named, you guessed it, for its inventors), and DSA, the Digital Signature Algorithm (invented by David Kravitz).

Because conventional cryptography was once the only available means for relaying secret information, the expense of secure channels and key distribution relegated its use only to those who could afford it, such as governments and large banks (or small children with secret decoder rings). Public key encryption is the technological revolution that provides strong cryptography to the adult masses. Remember the courier with the locked briefcase handcuffed to his wrist? Public-key encryption puts him out of business (probably to his relief).

How PGP works

PGP combines some of the best features of both conventional and public key cryptography. PGP is a *hybrid cryptosystem*.

When a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. Most cryptanalysis techniques exploit patterns found in the plaintext to crack the cipher. Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to cryptanalysis. (Files that are too short to compress or which don't compress well aren't compressed.)

PGP then creates a *session key*, which is a one-time-only secret key. This key is a random number generated from the random movements of your mouse and the keystrokes you type. This session key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is ciphertext. Once the data is encrypted, the session key is then encrypted to the recipient's public key. This public key-encrypted session key is transmitted along with the ciphertext to the recipient.

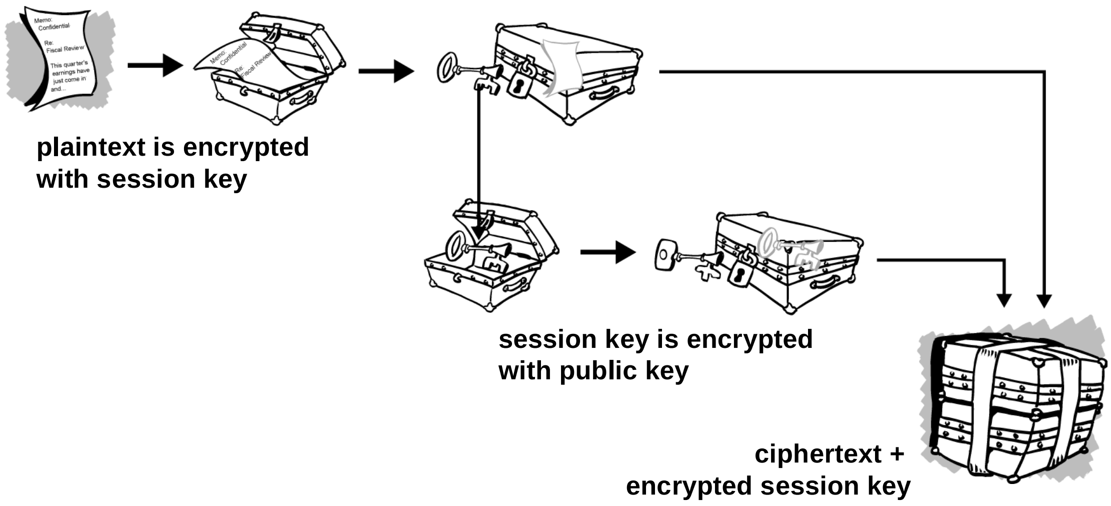
Memo: Confidential
Re: Fiscal Review
This quarter's earnings have just come in and...
plaintext is encrypted with session key
session key is encrypted with public key
ciphertext + encrypted session key

Figure 1-4. How PGP encryption works

Decryption works in the reverse. The recipient's copy of PGP uses his or her private key to recover the temporary session key, which PGP then uses to decrypt the conventionally-encrypted ciphertext.

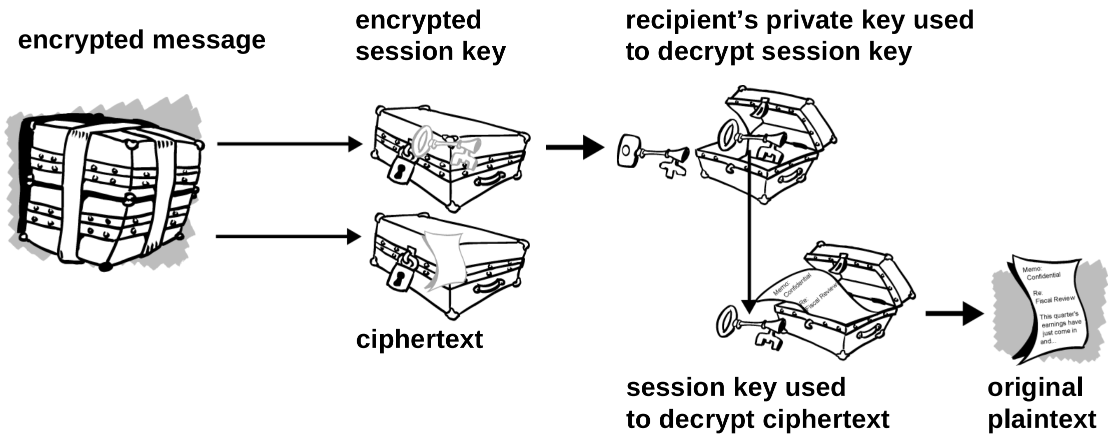
encrypted message
encrypted session key
recipient's private key used to decrypt session key
ciphertext
session key used to decrypt ciphertext
original plaintext

Figure 1-5. How PGP decryption works

The combination of the two encryption methods combines the convenience of public key encryption with the speed of conventional encryption. Conventional encryption is about 1,000 times faster than public key encryption. Public key encryption in turn provides a solution to key distribution and data transmission issues. Used together, performance and key distribution are improved without any sacrifice in security.

Keys

A key is a value that works with a cryptographic algorithm to produce a specific ciphertext. Keys are basically really, really, really big numbers. Key size is measured in bits; the number representing a 1024-bit key is darn huge. In public key cryptography, the bigger the key, the more secure the ciphertext.

However, public key size and conventional cryptography's secret key size are totally unrelated. A conventional 80-bit key has the equivalent strength of a 1024-bit public key. A conventional 128-bit key is equivalent to a 3000-bit public key. Again, the bigger the key, the more secure, but the algorithms used for each type of cryptography are very different and thus comparison is like that of apples to oranges.

While the public and private keys are mathematically related, it’s very difficult to derive the private key given only the public key; however, deriving the private key is always possible given enough time and computing power. This makes it very important to pick keys of the right size; large enough to be secure, but small enough to be applied fairly quickly. Additionally, you need to consider who might be trying to read your files, how determined they are, how much time they have, and what their resources might be.

Larger keys will be cryptographically secure for a longer period of time. If what you want to encrypt needs to be hidden for many years, you might want to use a very large key. Of course, who knows how long it will take to determine your key using tomorrow’s faster, more efficient computers? There was a time when a 56-bit symmetric key was considered extremely safe.

Keys are stored in encrypted form. PGP stores the keys in two files on your hard disk; one for public keys and one for private keys. These files are called *keyrings.* As you use PGP, you will typically add the public keys of your recipients to your public keyring. Your private keys are stored on your private keyring. If you lose your private keyring, you will be unable to decrypt any information encrypted to keys on that ring.

Digital signatures

A major benefit of public key cryptography is that it provides a method for employing *digital signatures*. Digital signatures enable the recipient of information to verify the authenticity of the information’s origin, and also verify that the information is intact. Thus, public key digital signatures provide *authentication* and data *integrity*. A digital signature also provides *non-repudiation*, which means that it prevents the sender from claiming that he or she did not actually send the information. These features are every bit as fundamental to cryptography as privacy, if not more.

A digital signature serves the same purpose as a handwritten signature. However, a handwritten signature is easy to counterfeit. A digital signature is superior to a handwritten signature in that it is nearly impossible to counterfeit, plus it attests to the contents of the information as well as to the identity of the signer.

Some people tend to use signatures more than they use encryption. For example, you may not care if anyone knows that you just deposited $1000 in your account, but you do want to be darn sure it was the bank teller you were dealing with.

The basic manner in which digital signatures are created is illustrated in *Figure 1-6*. Instead of encrypting information using someone else's public key, you encrypt it with your private key. If the information can be decrypted with your public key, then it must have originated with you.

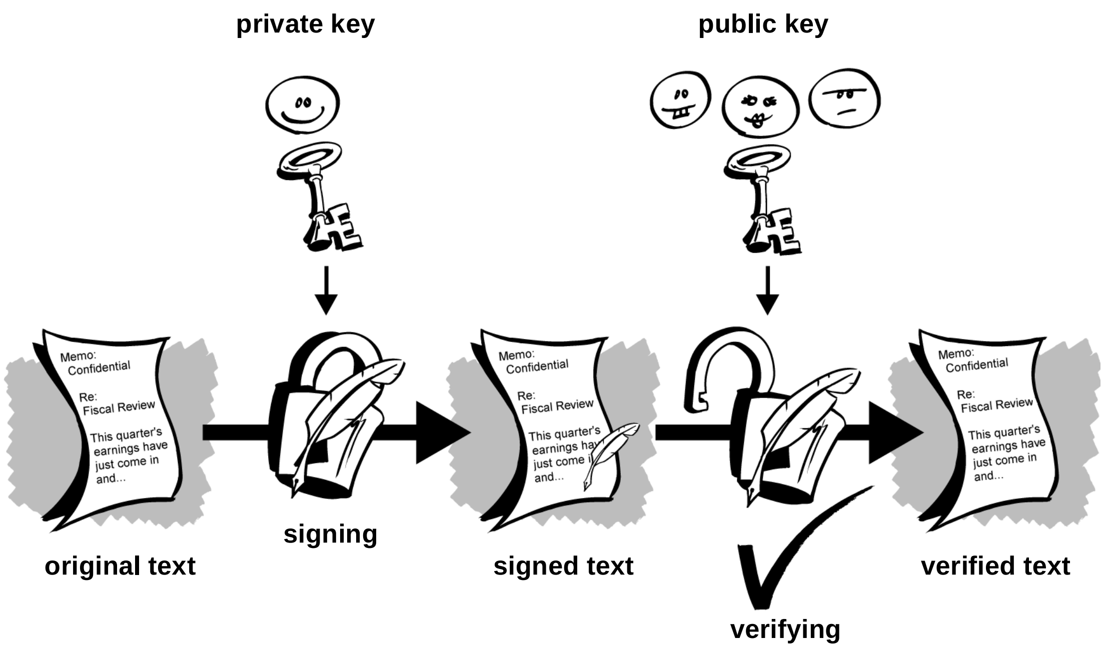

Figure 1-6. Simple digital signatures

Hash functions

The system described above has some problems. It is slow, and it produces an enormous volume of data—at least double the size of the original information. An improvement on the above scheme is the addition of a one-way *hash function* in the process. A one-way hash function takes variable-length input—in this case, a message of any length, even thousands or millions of bits—and produces a fixed-length output; say, 160-bits. The hash function ensures that, if the information is changed in any way—even by just one bit—an entirely different output value is produced.

PGP uses a cryptographically strong hash function on the plaintext the user is signing. This generates a fixed-length data item known as a *message digest*. (Again, any change to the information results in a totally different digest.)

Then PGP uses the digest and the private key to create the “signature.” PGP transmits the signature and the plaintext together. Upon receipt of the message, the recipient uses PGP to recompute the digest, thus verifying the signature. PGP can encrypt the plaintext or not; signing plaintext is useful if some of the recipients are not interested in or capable of verifying the signature.

As long as a secure hash function is used, there is no way to take someone's signature from one document and attach it to another, or to alter a signed message in any way. The slightest change in a signed document will cause the digital signature verification process to fail.

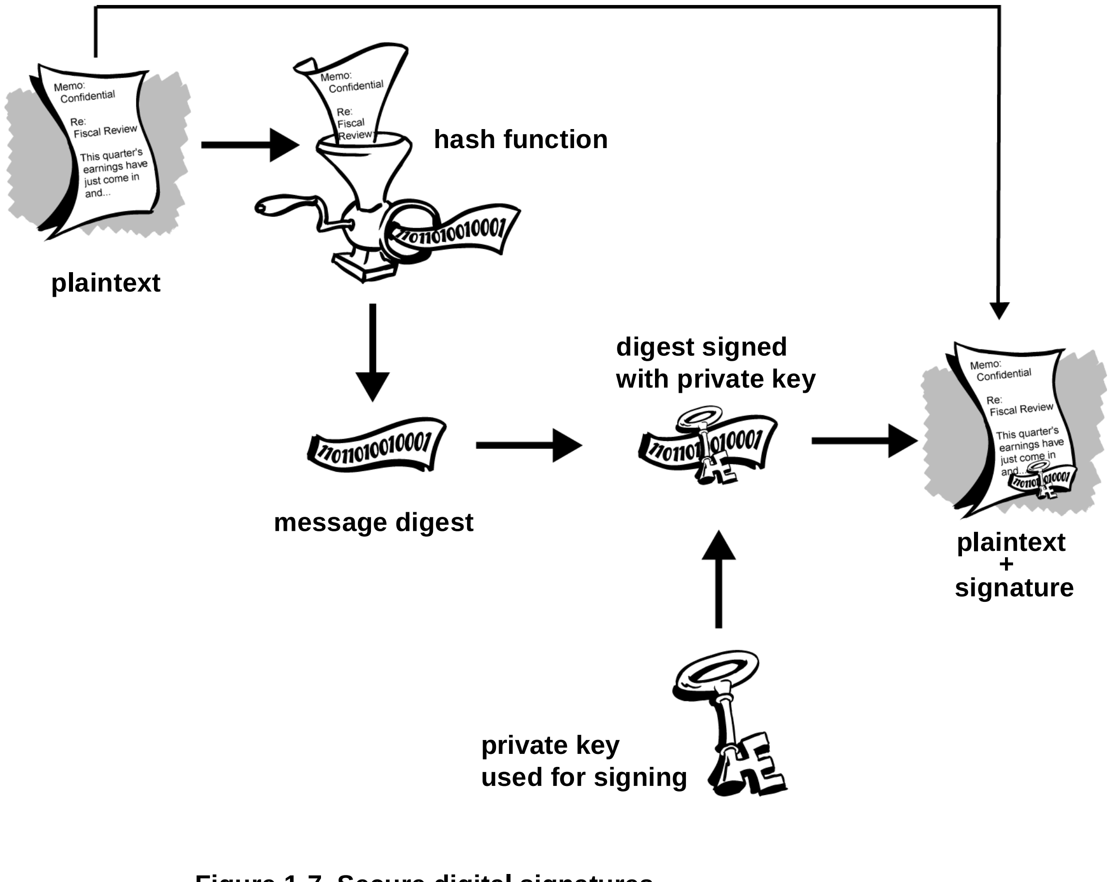
Memo: Confidential
Re: Fiscal Review
This quarter's earnings have just come in and...
plaintext
hash function
11011010010001
message digest
digest signed with private key
private key used for signing
plaintext + signature

Figure 1-7. Secure digital signatures

Digital signatures play a major role in authenticating and *validating* other PGP users’ keys.

Digital certificates

One issue with public key cryptosystems is that users must be constantly vigilant to ensure that they are encrypting to the correct person's key. In an environment where it is safe to freely exchange keys via public servers, *man-in-the-middle* attacks are a potential threat. In this type of attack, someone posts a phony key with the name and user ID of the user's intended recipient. Data encrypted to— and intercepted by—the true owner of this bogus key is now in the wrong hands.

In a public key environment, it is vital that you are assured that the public key to which you are encrypting data is in fact the public key of the intended recipient and not a forgery. You could simply encrypt only to those keys which have been physically handed to you. But suppose you need to exchange information with people you have never met; how can you tell that you have the correct key?

Digital certificates, or *certs*, simplify the task of establishing whether a public key truly belongs to the purported owner.

A certificate is a form of credential. Examples might be your driver's license, your social security card, or your birth certificate. Each of these has some information on it identifying you and some authorization stating that someone else has confirmed your identity. Some certificates, such as your passport, are important enough confirmation of your identity that you would not want to lose them, lest someone use them to impersonate you.

A digital certificate is data that functions much like a physical certificate. A digital certificate is information included with a person's public key that helps others verify that a key is genuine or *valid*. Digital certificates are used to thwart attempts to substitute one person's key for another.

A digital certificate consists of three things:

- A public key.
- Certificate information. ("Identity" information about the user, such as name, user ID, and so on.)
- One or more digital signatures.

The purpose of the digital signature on a certificate is to state that the certificate information has been attested to by some other person or entity. The digital signature does not attest to the authenticity of the certificate as a whole; it vouches only that the signed identity information goes along with, or *is bound to*, the public key.

Thus, a certificate is basically a public key with one or two forms of ID attached, plus a hearty stamp of approval from some other trusted individual.

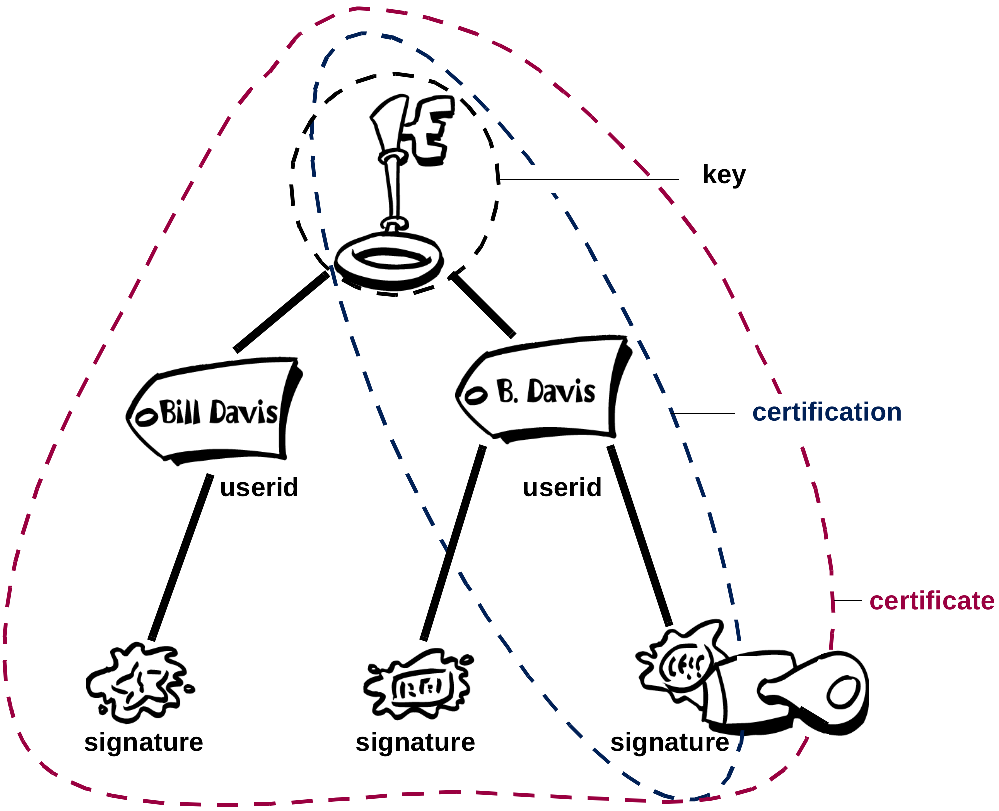

Figure 1-8. Anatomy of a PGP certificate

Certificate distribution

Certificates are utilized when it's necessary to exchange public keys with someone else. For small groups of people who wish to communicate securely, it is easy to manually exchange diskettes or emails containing each owner's public key. This is *manual public key distribution*, and it is practical only to a certain point. Beyond that point, it is necessary to put systems into place that can provide the necessary security, storage, and exchange mechanisms so coworkers, business partners, or strangers could communicate if need be. These can come in the form of storage-only repositories called *Certificate Servers*, or more structured systems that provide additional key management features and are called *Public Key Infrastructures (PKIs)*.

Certificate servers

A *certificate server*, also called a *cert server* or a *key server*, is a database that allows users to submit and retrieve digital certificates. A cert server usually provides some administrative features that enable a company to maintain its security policies—for example, allowing only those keys that meet certain requirements to be stored.

Public Key Infrastructures

A PKI contains the certificate storage facilities of a certificate server, but also provides certificate management facilities (the ability to issue, revoke, store, retrieve, and trust certificates). The main feature of a PKI is the introduction of what is known as a *Certification Authority*, or *CA*, which is a human entity—a person, group, department, company, or other association—that an organization has authorized to issue certificates to its computer users. (A CA's role is analogous to a country's government's Passport Office.) A CA creates certificates and digitally signs them using the CA's private key. Because of its role in creating certificates, the CA is the central component of a PKI. Using the CA's public key, anyone wanting to verify a certificate's authenticity verifies the issuing CA's digital signature, and hence, the integrity of the contents of the certificate (most importantly, the public key and the identity of the certificate holder).

Certificate formats

A digital certificate is basically a collection of identifying information bound together with a public key and signed by a trusted third party to prove its authenticity. A digital certificate can be one of a number of different *formats*.

PGP recognizes two different certificate formats:

- PGP certificates
- X.509 certificates

PGP certificate format

A PGP certificate includes (but is not limited to) the following information:

- **The PGP version number**—this identifies which version of PGP was used to create the key associated with the certificate.
- **The certificate holder's public key**—the public portion of your key pair, together with the algorithm of the key: RSA, DH (Diffie-Hellman), or DSA (Digital Signature Algorithm).

- **The certificate holder's information**—this consists of "identity" information about the user, such as his or her name, user ID, photograph, and so on.
- **The digital signature of the certificate owner**—also called a *self-signature*, this is the signature using the corresponding private key of the public key associated with the certificate.
- **The certificate's validity period**—the certificate's start date / time and expiration date / time; indicates when the certificate will expire.
- **The preferred symmetric encryption algorithm for the key**—indicates the encryption algorithm to which the certificate owner prefers to have information encrypted. The supported algorithms are CAST, IDEA or Triple-DES.

You might think of a PGP certificate as a public key with one or more labels tied to it (see Figure 1-9). On these 'labels' you'll find information identifying the owner of the key and a signature of the key's owner, which states that the key and the identification go together. (This particular signature is called a *self-signature*; every PGP certificate contains a self-signature.)

One unique aspect of the PGP certificate format is that a single certificate can contain multiple signatures. Several or many people may sign the key / identification pair to attest to their own assurance that the public key definitely belongs to the specified owner. If you look on a public certificate server, you may notice that certain certificates, such as that of PGP's creator, Phil Zimmermann, contain many signatures.

Some PGP certificates consist of a public key with several labels, each of which contains a different means of identifying the key's owner (for example, the owner's name and corporate email account, the owner's nickname and home email account, a photograph of the owner—all in one certificate). The list of signatures of each of those identities may differ; signatures attest to the authenticity that one of the labels belongs to the public key, not that all the labels on the key are authentic. (Note that 'authentic' is in the eye of its beholder—signatures are opinions, and different people devote different levels of due diligence in checking authenticity before signing a key.)

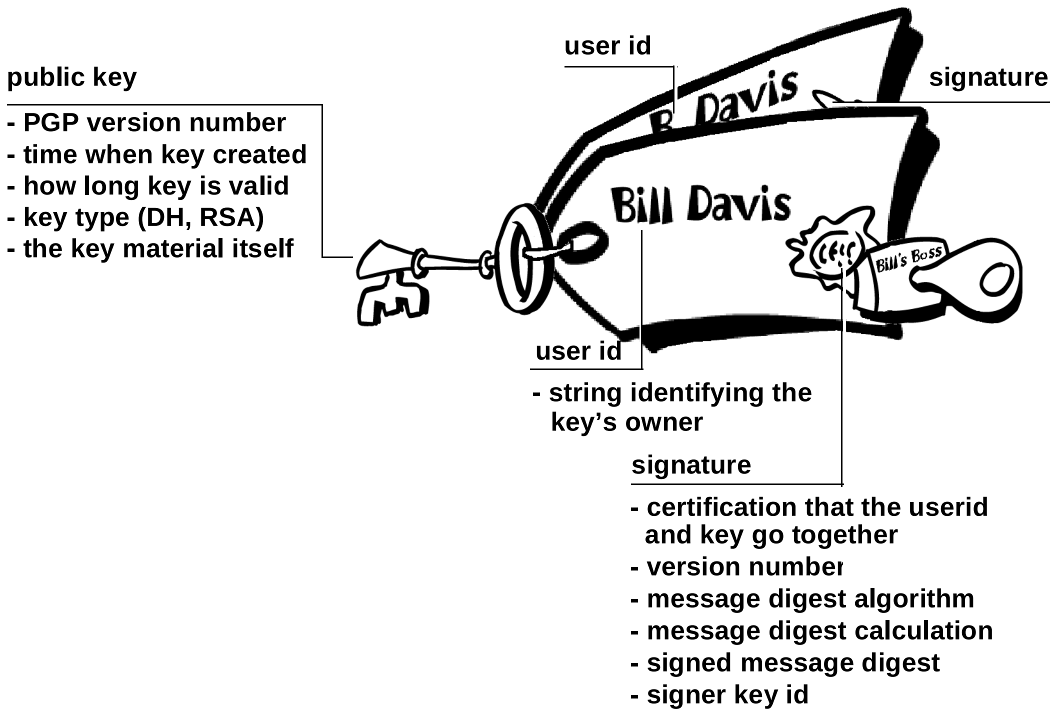

Figure 1-9. A PGP certificate

X.509 certificate format

X.509 is another very common certificate format. All X.509 certificates comply with the ITU-T X.509 international standard; thus (theoretically) X.509 certificates created for one application can be used by any application complying with X.509. In practice, however, different companies have created their own extensions to X.509 certificates, not all of which work together.

A certificate requires someone to validate that a public key and the name of the key's owner go together. With PGP certificates, anyone can play the role of validator. With X.509 certificates, the validator is always a Certification Authority or someone designated by a CA. (Bear in mind that PGP certificates also fully support a hierarchical structure using a *CA* to validate certificates.)

An X.509 certificate is a collection of a standard set of fields containing information about a user or device and their corresponding public key. The X.509 standard defines what information goes into the certificate, and describes how to encode it (the data format). All X.509 certificates have the following data:

- **The X.509 version number**—this identifies which version of the X.509 standard applies to this certificate, which affects what information can be specified in it. The most current is version 3.

- **The certificate holder's public key**—the public key of the certificate holder, together with an algorithm identifier which specifies which cryptosystem the key belongs to and any associated key parameters.

- **The serial number of the certificate**—the entity (application or person) that created the certificate is responsible for assigning it a unique serial number to distinguish it from other certificates it issues. This information is used in numerous ways; for example when a certificate is revoked, its serial number is placed in a *Certificate Revocation List* or *CRL*.

- **The certificate holder's unique identifier—** (or *DN—distinguished name*). This name is intended to be unique across the Internet. This name is intended to be unique across the Internet. A DN consists of multiple subsections and may look something like this:

  CN=Bob Allen, OU=Total Network Security Division, O=Network Associates, Inc., C=US

  (These refer to the subject's **C**ommon **N**ame, **O**rganizational **U**nit, **O**rganization, and **C**ountry.)

- **The certificate's validity period**—the certificate's start date/time and expiration date/time; indicates when the certificate will expire.

- **The unique name of the certificate issuer**—the unique name of the entity that signed the certificate. This is normally a CA. Using the certificate implies trusting the entity that signed this certificate. (Note that in some cases, such as *root* or *top-level* CA certificates, the issuer signs its own certificate.)

- **The digital signature of the issuer**—the signature using the private key of the entity that issued the certificate.

- **The signature algorithm identifier**—identifies the algorithm used by the CA to sign the certificate.

There are many differences between an X.509 certificate and a PGP certificate, but the most salient are as follows:

- you can create your own PGP certificate; you must request and be issued an X.509 certificate from a Certification Authority

- X.509 certificates natively support only a single name for the key's owner

- X.509 certificates support only a single digital signature to attest to the key's validity

To obtain an X.509 certificate, you must ask a CA to issue you a certificate. You provide your public key, proof that you possess the corresponding private key, and some specific information about yourself. You then digitally sign the information and send the whole package—the certificate *request*—to the CA. The CA then performs some due diligence in verifying that the information you provided is correct, and if so, generates the certificate and returns it.

You might think of an X.509 certificate as looking like a standard paper certificate (similar to one you might have received for completing a class in basic First Aid) with a public key taped to it. It has your name and some information about you on it, plus the signature of the person who issued it to you.

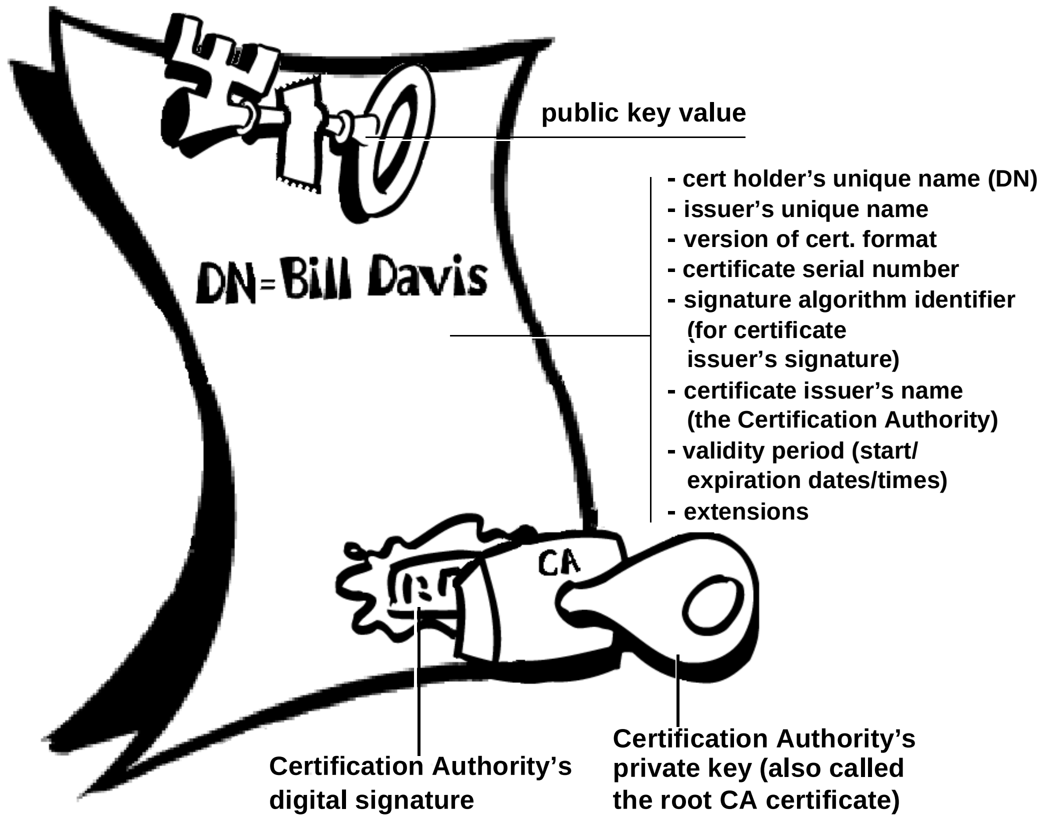
public key value
DN=Bill Davis
- cert holder's unique name (DN)
- issuer's unique name
- version of cert. format
- certificate serial number
- signature algorithm identifier (for certificate issuer's signature)
- certificate issuer's name (the Certification Authority)
- validity period (start/ expiration dates/times)
- extensions
CA
Certification Authority's digital signature
Certification Authority's private key (also called the root CA certificate)

Figure 1-10. An X.509 certificate

Probably the most widely visible use of X.509 certificates today is in web browsers.

Validity and trust

Every user in a public key system is vulnerable to mistaking a phony key (certificate) for a real one. *Validity* is confidence that a public key certificate belongs to its purported owner. Validity is essential in a public key environment where you must constantly establish whether or not a particular certificate is authentic.

When you've assured yourself that a certificate belonging to someone else is valid, you can sign the copy on your keyring to attest to the fact that you've checked the certificate and that it's an authentic one. If you want others to know that you gave the certificate your stamp of approval, you can export the signature to a certificate server so that others can see it.

As described in the section, "Public Key Infrastructures." some companies designate one or more Certification Authorities (CAs) to indicate certificate validity. In an organization using a PKI with X.509 certificates, it is the job of the CA to *issue* certificates to users—a process which generally entails responding to a user's request for a certificate. In an organization using PGP certificates without a PKI, it is the job of the CA to check the authenticity of all PGP certificates and then sign the good ones. Basically, the main purpose of a CA is to bind a public key to the identification information contained in the certificate and thus assure third parties that some measure of care was taken to ensure that this binding of the identification information and key is valid.

The CA is the Grand Pooh-bah of validation in an organization; someone whom everyone trusts, and in some organizations, like those using a PKI, no certificate is considered valid unless it has been signed by a trusted CA.

Checking validity

One way to establish validity is to go through some manual process. There are several ways to accomplish this. You could require your intended recipient to physically hand you a copy of his or her public key. But this is often inconvenient and inefficient.

Another way is to manually check the certificate's *fingerprint*. Just as every human's fingerprints are unique, every PGP certificate's fingerprint is unique. The fingerprint is a hash of the user's certificate and appears as one of the certificate's properties. In PGP, the fingerprint can appear as a hexadecimal number or a series of so-called *biometric words*, which are phonetically distinct and are used to make the fingerprint identification process a little easier.

You can check that a certificate is valid by calling the key's owner (so that you originate the transaction) and asking the owner to read his or her key's fingerprint to you and verifying that fingerprint against the one you believe to be the real one. This works if you know the owner's voice, but, how do you manually verify the identity of someone you don't know? Some people put the fingerprint of their key on their business cards for this very reason.

Another way to establish validity of someone's certificate is to *trust* that a third individual has gone through the process of validating it.

A CA, for example, is responsible for ensuring that prior to issuing to a certificate, he or she carefully checks it to be sure the public key portion really belongs to the purported owner. Anyone who trusts the CA will automatically consider any certificates signed by the CA to be valid.

Another aspect of checking validity is to ensure that the certificate has not been revoked. For more information, see the section, "Certificate Revocation".

Establishing trust

You validate *certificates*. You trust *people*. More specifically, you trust people to validate other people' certificates. Typically, unless the owner hands you the certificate, you have to go by someone else's word that it is valid.

Meta and trusted introducers

In most situations, people completely trust the CA to establish certificates' validity. This means that everyone else relies upon the CA to go through the whole manual validation process for them. This is fine up to a certain number of users or number of work sites, and then it is not possible for the CA to maintain the same level of quality validation. In that case, adding other validators to the system is necessary.

A CA can also be a *meta-introducer*. A meta-introducer bestows not only validity on keys, but bestows the *ability to trust keys* upon others. Similar to the king who hands his seal to his trusted advisors so they can act on his authority, the meta-introducer enables others to act as *trusted introducers*. These trusted introducers can validate keys to the same effect as that of the meta-introducer. They cannot, however, create new trusted introducers.

Meta-introducer and trusted introducer are PGP terms. In an X.509 environment, the meta-introducer is called the *root Certification Authority* (*root CA*) and trusted introducers *subordinate* Certification Authorities.

The root CA uses the private key associated with a special certificate type called a *root CA certificate* to sign certificates. Any certificate signed by the root CA certificate is viewed as valid by any other certificate signed by the root. This validation process works even for certificates signed by other CAs in the system—as long as the root CA certificate signed the subordinate CA's certificate, any certificate signed by the CA is considered valid to others within the hierarchy. This process of checking back up through the system to see who signed whose certificate is called tracing a *certification path* or *certification chain*.

Trust models

In relatively closed systems, such as within a small company, it is easy to trace a certification path back to the root CA. However, users must often communicate with people outside of their corporate environment, including some whom they have never met, such as vendors, customers, clients, associates, and so on. Establishing a line of trust to those who have not been explicitly trusted by your CA is difficult.

Companies follow one or another *trust model*, which dictates how users will go about establishing certificate validity. There are three different models:

- Direct Trust
- Hierarchical Trust
- A Web of Trust

Direct Trust

Direct trust is the simplest trust model. In this model, a user trusts that a key is valid because he or she knows where it came from. All cryptosystems use this form of trust in some way. For example, in web browsers, the root Certification Authority keys are directly trusted because they were shipped by the manufacturer. If there is any form of hierarchy, it extends from these directly trusted certificates.

In PGP, a user who validates keys herself and never sets another certificate to be a trusted introducer is using direct trust.

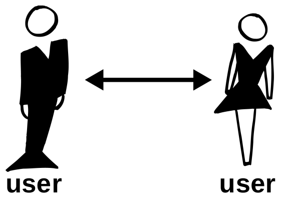

Figure 1-11. Direct trust

Hierarchical Trust

In a hierarchical system, there are a number of “root” certificates from which trust extends. These certificates may certify certificates themselves, or they may certify certificates that certify still other certificates down some chain. Consider it as a big trust “tree.” The “leaf” certificate's validity is verified by tracing backward from its certifier, to other certifiers, until a directly trusted root certificate is found.

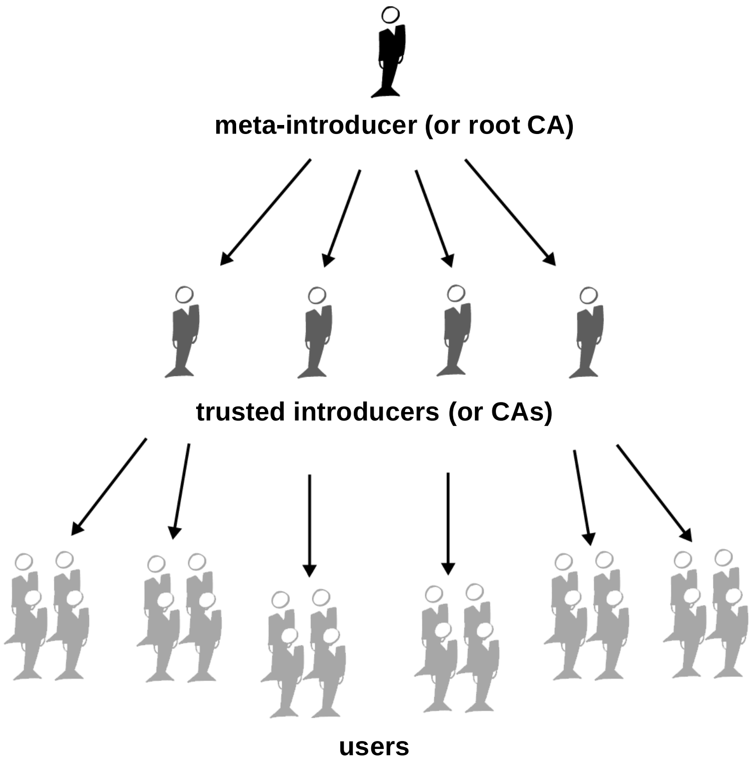

Figure 1-12. Hierarchical trust

Web of Trust

A web of trust encompasses both of the other models, but also adds the notion that trust is in the eye of the beholder (which is the real-world view) and the idea that more information is better. It is thus a cumulative trust model. A certificate might be trusted directly, or trusted in some chain going back to a directly trusted root certificate (the meta-introducer), or by some group of introducers.

Perhaps you've heard of the term *six degrees of separation*, which suggests that any person in the world can determine some link to any other person in the world using six or fewer other people as intermediaries. This is a web of introducers.

It is also the PGP view of trust. PGP uses digital signatures as its form of introduction. When any user signs another's key, he or she becomes an introducer of that key. As this process goes on, it establishes a *web of trust.*

In a PGP environment, *any* user can act as a certifying authority. Any PGP user can validate another PGP user's public key certificate. However, such a certificate is only valid to another user if the relying party recognizes the validator as a trusted introducer. (That is, you trust my opinion that others' keys are valid only if you consider me to be a trusted introducer. Otherwise, my opinion on other keys' validity is moot.)

Stored on each user's public keyring are indicators of

- whether or not the user considers a particular key to be valid
- the level of trust the user places on the key that the key's owner can serve as certifier of others' keys

You indicate, on your copy of my key, whether you think my judgement counts. It's really a reputation system: certain people are reputed to give good signatures, and people trust them to attest to other keys' validity.

Levels of trust in PGP

The highest level of trust in a key, *implicit* trust, is trust in your own key pair. PGP assumes that if you own the private key, you must trust the actions of its related public key. Any keys signed by your implicitly trusted key are valid.

There are three levels of trust you can assign to someone else's public key:

- *Complete* trust
- *Marginal* trust
- No trust (or *Untrusted*)

To make things confusing, there are also three levels of validity:

- Valid
- Marginally valid
- Invalid

To define another's key as a trusted introducer, you

1. Start with a valid key, one that is either

- signed by you or
- signed by another trusted introducer

and then

2. Set the level of trust you feel the key’s owner is entitled.

For example, suppose your key ring contains Alice’s key. You have validated Alice’s key and you indicate this by signing it. You know that Alice is a real stickler for validating others’ keys. You therefore assign her key with Complete trust. This makes Alice a Certification Authority. If Alice signs another’s key, it appears as Valid on your keyring.

PGP requires one Completely trusted signature or two Marginally trusted signatures to establish a key as valid. PGP’s method of considering two Marginals equal to one Complete is similar to a merchant asking for two forms of ID. You might consider Alice fairly trustworthy and also consider Bob fairly trustworthy. Either one alone runs the risk of accidentally signing a counterfeit key, so you might not place complete trust in either one. However, the odds that both individuals signed the same phony key are probably small.

Certificate Revocation

Certificates are only useful while they are valid. It is unsafe to simply assume that a certificate is valid forever. In most organizations and in all PKIs, certificates have a restricted lifetime. This constrains the period in which a system is vulnerable should a certificate compromise occur.

Certificates are thus created with a scheduled *validity period*: a start date / time and an expiration date / time. The certificate is expected to be usable for its entire validity period (its *lifetime*). When the certificate expires, it will no longer be valid, as the authenticity of its key / identification pair are no longer assured. (The certificate can still be safely used to reconfirm information that was encrypted or signed within the validity period—it should not be trusted for cryptographic tasks moving forward, however.)

There are also situations where it is necessary to invalidate a certificate prior to its expiration date, such as when an the certificate holder terminates employment with the company or suspects that the certificate’s corresponding private key has been compromised. This is called *revocation*. A revoked certificate is *much* more suspect than an expired certificate. Expired certificates are unusable, but do not carry the same threat of compromise as a revoked certificate.

Anyone who has signed a certificate can revoke his or her signature on the certificate (provided he or she uses the same private key that created the signature). A revoked signature indicates that the signer no longer believes the public key and identification information belong together, or that the certificate's public key (or corresponding private key) has been compromised. A revoked signature should carry nearly as much weight as a revoked certificate.

With X.509 certificates, a revoked signature is practically the same as a revoked certificate given that the only signature on the certificate is the one that made it valid in the first place—the signature of the CA. PGP certificates provide the added feature that you can revoke your entire certificate (not just the signatures on it) if you yourself feel that the certificate has been compromised.

Only the certificate's owner (the holder of its corresponding private key) or someone whom the certificate's owner has *designated* as a revoker can revoke a PGP certificate. (Designating a revoker is a useful practice, as it's often the loss of the passphrase for the certificate's corresponding private key that leads a PGP user to revoke his or her certificate—a task that is only possible if one has access to the private key.) Only the certificate's issuer can revoke an X.509 certificate.

Communicating that a certificate has been revoked

When a certificate is revoked, it is important to make potential users of the certificate aware that it is no longer valid. With PGP certificates, the most common way to communicate that a certificate has been revoked is to post it on a certificate server so others who may wish to communicate with you are warned not to use that public key.

In a PKI environment, communication of revoked certificates is most commonly achieved via a data structure called a *Certificate Revocation List*, or *CRL*, which is published by the CA. The CRL contains a time-stamped, validated list of all revoked, unexpired certificates in the system. Revoked certificates remain on the list only until they expire, then they are removed from the list—this keeps the list from getting too long.

The CA distributes the CRL to users at some regularly scheduled interval (and potentially off-cycle, whenever a certificate is revoked). Theoretically, this will prevent users from unwittingly using a compromised certificate. It is possible, though, that there may be a time period between CRLs in which a newly compromised certificate is used.

What is a passphrase?

Most people are familiar with restricting access to computer systems via a *password*, which is a unique string of characters that a user types in as an identification code.

A *passphrase* is a longer version of a password, and in theory, a more secure one. Typically composed of multiple words, a passphrase is more secure against standard *dictionary attacks*, wherein the attacker tries all the words in the dictionary in an attempt to determine your password. The best passphrases are relatively long and complex and contain a combination of upper and lowercase letters, numeric and punctuation characters.

PGP uses a passphrase to encrypt your private key on your machine. Your private key is encrypted on your disk using a hash of your passphrase as the secret key. You use the passphrase to decrypt and use your private key. A passphrase should be hard for you to forget and difficult for others to guess. It should be something already firmly embedded in your long-term memory, rather than something you make up from scratch. Why? Because **if you forget your passphrase, you are out of luck**. Your private key is totally and absolutely useless without your passphrase and nothing can be done about it. Remember the quote earlier in this chapter? PGP is cryptography that will keep major governments out of your files. It will certainly keep you out of your files, too. Keep that in mind when you decide to change your passphrase to the punchline of that joke you can never quite remember.

Key splitting

They say that a secret is not a secret if it is known to more than one person. Sharing a private key pair poses such a problem. While it is not a recommended practice, sharing a private key pair is necessary at times. *Corporate Signing Keys*, for example, are private keys used by a company to sign—for example—legal documents, sensitive personnel information, or press releases to authenticate their origin. In such a case, it is worthwhile for multiple members of the company to have access to the private key. However, this means that any single individual can act fully on behalf of the company.

In such a case it is wise to *split* the key among multiple people in such a way that more than one or two people must present a piece of the key in order to reconstitute it to a usable condition. If too few pieces of the key are available, then the key is unusable.

Some examples are to split a key into three pieces and require two of them to reconstitute the key, or split it into two pieces and require both pieces. If a secure network connection is used during the reconstitution process, the key's shareholders need not be physically present in order to rejoin the key.

Technical details

This chapter provided a high-level introduction to cryptographic concepts and terminology. In Chapter 2, Phil Zimmermann, the creator of PGP, provides a more in-depth discussion of privacy, the technical details of how PGP works, including the various algorithms it uses, as well as various attacks and how to protect yourself against them.

For more information on cryptography, please refer to some of the books listed in the ”Related reading” section of the Preface.

Glossary

A5	a trade-secret cryptographic algorithm used in European cellular telephones.
Access control	a method of restricting access to resources, allowing only privileged entities access.
Additional recipient request key	a special key whose presence indicates that all messages encrypted to its associated base key should also be automatically encrypted to it. Sometimes referred to by its marketing term, *additional decryption key.*
AES (Advanced Encryption Standard)	NIST approved standards, usually used for the next 20 - 30 years.
AKEP (Authentication Key Exchange Protocol)	key transport based on symmetric encryption allowing two parties to exchange a shared secret key, secure against passive adversaries.
Algorithm (encryption)	a set of mathematical rules (logic) used in the processes of encryption and decryption.
Algorithm (hash)	a set of mathematical rules (logic) used in the processes of message digest creation and key/signature generation.
Anonymity	of unknown or undeclared origin or authorship, concealing an entity's identification.
ANSI (American National Standards Institute)	develops standards through various Accredited Standards Committees (ASC). The X9 committee focuses on security standards for the financial services industry.
API (Application Programming Interface)	provides the means to take advantage of software features, allowing dissimilar software products to interact upon one another.

ASN.1 (Abstract Syntax Notation One)	ISO/IEC standard for encoding rules used in ANSI X.509 certificates, two types exist - DER (Distinguished Encoding Rules) and BER (Basic Encoding Rules).
Asymmetric keys	a separate but integrated user key-pair, comprised of one public key and one private key. Each key is one way, meaning that a key used to encrypt information can not be used to decrypt the same data.
Authentication	to prove genuine by corroboration of the identity of an entity.
Authorization certificate	an electronic document to prove one's access or privilege rights, also to prove one is who they say they are.
Authorization	to convey official sanction, access or legal power to an entity.
Blind signature	ability to sign documents without knowledge of content, similar to a notary public.
Block cipher	a symmetric cipher operating on blocks of plain text and cipher text, usually 64 bits.
Blowfish	a 64-bit block symmetric cipher consisting of key expansion and data encryption. A fast, simple, and compact algorithm in the public domain written by Bruce Schneier.
CA (Certificate Authority)	a trusted third party (TTP) who creates certificates that consist of assertions on various attributes and binds them to an entity and/or to their public key.
CAPI (Crypto API)	Microsoft's crypto API for Windows-based operating systems and applications.
Capstone	an NSA-developed cryptographic chip that implements a US government Key Escrow capability.
CAST	a 64-bit block cipher using 64-bit key, six S-boxes with 8-bit input and 32-bit output, developed in Canada by Carlisle Adams and Stafford Tavares.

CBC (Cipher Block Chaining)	the process of having plain text XORed with the previous cipher text block before it is encrypted, thus adding a feedback mechanism to a block cipher.
CDK (Crypto Developer Kit)	a documented environment, including an API for third parties to write secure applications using a specific vendor's cryptographic library.
CERT (Computer Emergency Response Team)	security clearinghouse that promotes security awareness. CERT provides 24-hour technical assistance for computer and network security incidents. CERT is located at the Software Engineering Institute at Carnegie Mellon University in Pittsburgh, PA.
Certificate (digital certificate)	an electronic document attached to a public key by a trusted third party, which provides proof that the public key belongs to a legitimate owner and has not been compromised.
CFM (Cipher Feedback Mode)	a block cipher that has been implemented as a self-synchronizing stream cipher.
CDSA (Common Data Security Architecture)	Intel Architecture Labs (IAL) developed this framework to address the data security problems inherent to Internet and Intranet for use in Intel and others' Internet products.
Certification	endorsement of information by a trusted entity.
CHAP (Challenge Authentication Protocol)	a session-based, two-way password authentication scheme.
Cipher text	the result of manipulating either characters or bits via substitution, transposition, or both.
Clear text	characters in a human readable form or bits in a machine-readable form (also called *plain text*).
Confidentiality	the act of keeping something private and secret from all but those who are authorized to see it.

Cookie	Persistent Client State HTTP Cookie - a file or token of sorts, that is passed from the web server to the web client (your browser) that is used to identify you and could record personal information such as ID and password, mailing address, credit card number, and other information.
CRAB	a 1024-byte block cipher (similar to MD5), using techniques from a one-way hash function, developed by Burt Kaliski and Matt Robshaw at RSA Laboratories.
Credentials	something that provides a basis for credit or confidence.
CRL (Certificate Revocation List)	an online, up-to-date list of previously issued certificates that are no longer valid.
Cross-certification	two or more organizations or Certificate Authorities that share some level of trust.
Cryptanalysis	the art or science of transferring cipher text into plain text without initial knowledge of the key used to encrypt the plain text.
CRYPTOKI	same as PKCS #11.
Cryptography	the art and science of creating messages that have some combination of being private, signed, unmodified with non-repudiation.
Cryptosystem	a system comprised of cryptographic algorithms, all possible plain text, cipher text, and keys.
Data integrity	a method of ensuring information has not been altered by unauthorized or unknown means.
Decryption	the process of turning cipher text back into plain text.
DES (Data Encryption Standard)	a 64-bit block cipher, symmetric algorithm also known as Data Encryption Algorithm (DEA) by ANSI and DEA-1 by ISO. Widely used for over 20 years, adopted in 1976 as FIPS 46.

Dictionary attack	a calculated brute force attack to reveal a password by trying obvious and logical combinations of words.
Diffie-Hellman	the first public key algorithm, invented in 1976, using discrete logarithms in a finite field.
Digital cash	electronic money that is stored and transferred through a variety of complex protocols.
Direct trust	an establishment of peer-to-peer confidence.
Discrete logarithm	the underlying mathematical problem used in/by asymmetric algorithms, like Diffie-Hellman and Elliptic Curve. It is the inverse problem of modular exponentiation, which is a one-way function.
DMS (Defense Messaging System)	standards designed by the U.S. Department of Defense to provide a secure and reliable enterprise-wide messaging infrastructure for government and military agencies.
DNSSEC (Domain Name System Security Working Group)	a proposed *IETF* draft that will specify enhancements to the DNS protocol to protect the DNS against unauthorized modification of data and against masquerading of data origin. It will add data integrity and authentication capabilities to the DNS via digital signatures.
DSA (Digital Signature Algorithm)	a public key digital signature algorithm proposed by NIST for use in DSS.
Digital signature	an electronic identification of a person or thing created by using a public key algorithm. Intended to verify to a recipient the integrity of data and identity of the sender of the data.
DSS (Digital Signature Standard)	a NIST proposed standard (FIPS) for digital signatures using DSA.
ECC (Elliptic Curve Cryptosystem)	a unique method for creating public key algorithms based on mathematical curves over finite fields or with large prime numbers.

EDI (Electronic Data Interchange)	the direct, standardized computer-to-computer exchange of business documents (purchase orders, invoices, payments, inventory analyses, and others) between your organization and your suppliers and customers.
EES (Escrowed Encryption Standard)	a proposed U.S. government standard for escrowing private keys.
Elgamal scheme	used for both digital signatures and encryption based on discrete logarithms in a finite field; can be used with the DSA function.
Encryption	the process of disguising a message in such a way as to hide its substance.
Entropy	a mathematical measurement of the amount of uncertainty or randomness.
FEAL	a block cipher using 64-bit block and 64-bit key, design by A. Shimizu and S. Miyaguchi at NTT Japan.
Filter	a function, set of functions, or combination of functions that applies some number of transforms to its input set, yielding an output set containing only those members of the input set that satisfy the transform criteria. The selected members may or may not be further transformed in the resultant output set. An example would be a search function that accepts multiple strings having a boolean relationship `((like` *`a`* `or like` *`b`* `) but not containing` *`c`*`)`, and optionally forces the case of the found strings in the resultant output.
Fingerprint	a unique identifier for a key that is obtained by hashing specific portions of the key data.
FIPS (Federal Information Processing Standard)	a U.S. government standard published by NIST.
Firewall	a combination of hardware and software that protects the perimeter of the public/private network against certain attacks to ensure some degree of security.

GAK (Government Access to Keys)	a method for the government to escrow individual's private key.
Gost	a 64-bit symmetric block cipher using a 256-bit key, developed in the former Soviet Union.
GSS-API (Generic Security Services API)	a high-level security API based upon IETF RFC 1508, which isolates session-oriented application code from implementation details.
Hash function	a one-way hash function - a function that produces a message digest that cannot be reversed to produced the original.
HMAC	a key-dependent one-way hash function specifically intended for use with MAC (Message Authentication Code), and based upon IETF RFC 2104.
Hierarchical trust	a graded series of entities that distribute trust in an organized fashion, commonly used in ANSI X.509 issuing certifying authorities.
HTTP (HyperText Transfer Protocol)	a common protocol used to transfer documents between servers or from a server to a client.
IDEA (International Data Encryption Standard)	a 64-bit block symmetric cipher using 128-bit keys based on mixing operations from different algebraic groups. Considered one of the strongest algorithms.
IETF (Internet Engineering Task Force)	a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual.
Identity certificate	a signed statement that binds a key to the name of an individual and has the intended meaning of delegating authority from that named individual to the public key.
Initialization vector (IV)	a block of arbitrary data that serves as the starting point for a block cipher using a chaining feedback mode (see cipher block chaining).

Integrity	assurance that data is not modified (by unauthorized persons) during storage or transmittal.
IPSec	a TCP/IP layer encryption scheme under consideration within the IETF.
ISA/KMP (Internet Security Association, Key Mgt. Protocol)	defines the procedures for authenticating a communicating peer, creation and management of Security Associations, key generation techniques, and threat mitigation, for example, denial of service and replay attacks.
ISO (International Organization for Standardization)	responsible for a wide range of standards, like the OSI model and international relationship with ANSI on X.509.
ITU-T (International Telecommunication Union-Telecommunication)	formally the CCITT (Consultative Committee for International Telegraph and Telephone), a worldwide telecommunications technology standards organization.
Kerberos	a trusted third-party authentication protocol developed at MIT.
Key	a means of gaining or preventing access, possession, or control represented by any one of a large number of values.
Key escrow/recovery	a mechanism that allows a third party to retrieve the cryptographic keys used for data confidentiality, with the ultimate goal of recovery of encrypted data.
Key exchange	a scheme for two or more nodes to transfer a secret session key across an unsecured channel.
Key length	the number of bits representing the key size; the longer the key, the stronger it is.
Key management	the process and procedure for safely storing and distributing accurate cryptographic keys; the overall process of generating and distributing cryptographic key to authorized recipients in a secure manner.

Key splitting	a process for dividing portions of a single key between multiple parties, none having the ability to reconstruct the whole key.
LDAP (Lightweight Directory Access Protocol)	a simple protocol that supports access and search operations on directories containing information such as names, phone numbers, and addresses across otherwise incompatible systems over the Internet.
Lexical section	a distinct portion of a message that contains a specific class of data, for example, clear-signed data, encrypted data, and key data.
MAA (Message Authenticator Algorithm)	an ISO standard that produces a 32-bit hash, designed for IBM mainframes.
MAC (Message Authentication Code)	a key-dependent one-way hash function, requiring the use of the identical key to verify the hash.
MD2 (Message Digest 2)	128-bit one-way hash function designed by Ron Rivest, dependent on a random permutation of bytes.
MD4 (Message Digest 4)	128-bit one-way hash function designed by Ron Rivest, using a simple set of bit manipulations on 32-bit operands.
MD5 (Message Digest 5)	improved, more complex version of MD4, but still a 128-bit one-way hash function.
Message digest	a number that is derived from a message. Change a single character in the message and the message will have a different message digest.
MIC (Message Integrity Check)	originally defined in PEM for authentication using MD2 or MD5. Micalg (message integrity calculation) is used in secure MIME implementations.
MIME (Multipurpose Internet Mail Extensions)	a freely available set of specifications that offers a way to interchange text in languages with different character sets, and multimedia email among many different computer systems that use Internet mail standards.

MMB (Modular Multiplication-based Block)	based on IDEA, Joan Daemen developed this 128-bit key / 128-bit block size symmetric algorithm, not used because of its susceptibility to linear cryptanalysis.
MOSS (MIME Object Security Service)	defined in RFC 1848, it facilitates encryption and signature services for MIME, including key management based on asymmetric techniques (not widely used).
MSP (Message Security Protocol)	the military equivalent of PEM, an X.400-compatible application level protocol for securing e-mail, developed by the NSA in late 1980.
MTI	a one-pass key agreement protocol by Matsumoto, Takashima, and Imai that provides mutual key authentication without key confirmation or entity authentication.
NAT (Network Address Translator)	RFC 1631, a router connecting two networks together; one designated as inside, is addressed with either private or obsolete addresses that need to be converted into legal addresses before packets are forwarded onto the other network (designated as outside).
NIST (National Institute for Standards and Technology)	a division of the U.S. Dept. of Commerce that publishes open, interoperability standards called FIPS.
Non-repudiation	preventing the denial of previous commitments or actions.
Oakely	the “Oakley Session Key Exchange” provides a hybrid Diffie-Hellman session key exchange for use within the ISA / KMP framework. Oakley provides the important property of “Perfect Forward Secrecy.”
One-time pad	a large non-repeating set of truly random key letters used for encryption, considered the only perfect encryption scheme, invented by Major J. Mauborgne and G. Vernam in 1917.

One-way hash	a function of a variable string to create a fixed length value representing the original pre-image, also called message digest, fingerprint, message integrity check (MIC).
Orange Book	the National Computer Security Center book entitled *Department of Defense Trusted Computer Systems Evaluation Criteria* that defines security requirements.
PAP (Password Authentication Protocol)	an authentication protocol that allows PPP peers to authenticate one another, does not prevent unauthorized access but merely identifies the remote end.
Passphrase	an easy-to-remember phrase used for better security than a single password; key crunching converts it into a random key.
Password	a sequence of characters or a word that a subject submits to a system for purposes of authentication, validation, or verification.
PCT (Private Communication Technology)	a protocol developed by Microsoft and Visa for secure communications on the Internet.
PEM (Privacy Enhanced Mail)	a protocol to provide secure internet mail, (RFC 1421-1424) including services for encryption, authentication, message integrity, and key management. PEM uses ANSI X.509 certificates.
Perfect forward secrecy	a cryptosystem in which the cipher text yields no possible information about the plain text, except possibly the length.
Primitive filter	a function that applies a single transform to its input set, yielding an output set containing only those members of the input set that satisfy the transform criteria. An example would be a search function that accepts only a single string and outputs a list of line numbers where the string was found.

Pretty Good Privacy (PGP)	an application and protocol (RFC 1991) for secure e-mail and file encryption developed by Phil R. Zimmermann. Originally published as Freeware, the source code has always been available for public scrutiny. PGP uses a variety of algorithms, like IDEA, RSA, DSA, MD5, SHA-1 for providing encryption, authentication, message integrity, and key management. PGP is based on the "Web-of-Trust" model and has worldwide deployment.
PGP/MIME	an IETF standard (RFC 2015) that provides privacy and authentication using the Multipurpose Internet Mail Extensions (MIME) security content types described in RFC1847, currently deployed in PGP 5.0 and later versions.
PKCS (Public Key Crypto Standards)	a set of *de facto* standards for public key cryptography developed in cooperation with an informal consortium (Apple, DEC, Lotus, Microsoft, MIT, RSA, and Sun) that includes algorithm-specific and algorithm-independent implementation standards. Specifications defining message syntax and other protocols controlled by RSA Data Security Inc.
PKI (Public Key Infrastructure)	a widely available and accessible certificate system for obtaining an entity's public key with some degree of certainty that you have the "right" key and that it has not been revoked.
Plain text (or clear text)	the human readable data or message before it is encrypted.
Pseudo-random number	a number that results from applying randomizing algorithms to input derived from the computing environment, for example, mouse coordinates. See *random number*.
Private key	the privately held "secret" component of an integrated asymmetric key pair, often referred to as the decryption key.

Public key	the publicly available component of an integrated asymmetric key pair often referred to as the encryption key.
RADIUS (Remote Authentication Dial-In User Service)	an IETF protocol (developed by Livingston, Enterprise), for distributed security that secures remote access to networks and network services against unauthorized access. RADIUS consists of two pieces - authentication server code and client protocols.
Random number	an important aspect to many cryptosystems, and a necessary element in generating a unique key(s) that are unpredictable to an adversary. True random numbers are usually derived from analog sources, and usually involve the use of special hardware.
RC2 (Rivest Cipher 2)	variable key size, 64-bit block symmetric cipher, a trade secret held by RSA, SDI.
RC4 (Rivest Cipher 4)	variable key size stream cipher, once a proprietary algorithm of RSA Data Security, Inc.
RC5 (Rivest Cipher 5)	a block cipher with a variety of arguments, block size, key size, and number of rounds.
RIPE-MD	an algorithm developed for the European Community’s RIPE project, designed to resist known cryptanalysis attacks and produce a 128-bit hash value, a variation of MD4.
REDOC	a U.S.-patented block cipher algorithm developed by M. Wood, using a 160-bit key and an 80-bit block.
Revocation	retraction of certification or authorization.
RFC (Request for Comment)	an IETF document, either FYI (For Your Information) RFC sub-series that are overviews and introductory or STD RFC sub-series that identify specify Internet standards. Each RFC has an RFC number by which it is indexed and by which it can be retrieved (www.ietf.org).

ROT-13 (Rotation Cipher)	a simple substitution (Caesar) cipher, rotating each 26 letters 13 places.
RSA	short for RSA Data Security, Inc.; or referring to the principals - Ron Rivest, Adi Shamir, and Len Adleman; or referring to the algorithm they invented. The RSA algorithm is used in public key cryptography and is based on the fact that it is easy to multiply two large prime numbers together, but hard to factor them out of the product.
SAFER (Secure And Fast Encryption Routine)	a non-proprietary block cipher 64-bit key encryption algorithm. It is not patented, is available license free, and was developed by Massey, who also developed IDEA.
Salt	a random string that is concatenated with passwords (or random numbers) before being operated on by a one-way function. This concatenation effectively lengthens and obscures the password, making the cipher text less susceptible to dictionary attacks.
SDSI (Simple Distributed Security Infrastructure)	a new *PKI* proposal from Ronald L. Rivest (MIT), and Butler Lampson (Microsoft). It provides a means of defining groups and issuing group-membership, access-control lists, and security policies. SDSI's design emphasizes linked local name spaces rather than a hierarchical global name space.
SEAL (Software-optimized Encryption ALgorithm)	a fast stream cipher for 32-bit machines designed by Rogaway and Coppersmith.
Secret key	either the “private key” in public key (asymmetric) algorithms or the “session key” in symmetric algorithms.
Secure channel	a means of conveying information from one entity to another such that an adversary does not have the ability to reorder, delete, insert, or read (SSL, IPSec, whispering in someone’s ear).

Self-signed key	a public key that has been signed by the corresponding private key for proof of ownership.
SEPP (Secure Electronic Payment Protocol)	an open specification for secure bankcard transactions over the Internet. Developed by IBM, Netscape, GTE, Cybercash, and MasterCard.
SESAME (Secure European System for Applications in a Multi-vendor Environment)	European research and development project that extended Kerbros by adding authorization and access services.
Session key	the secret (symmetric) key used to encrypt each set of data on a transaction basis. A different session key is used for each communication session.
SET (Secure Electronic Transaction)	provides for secure exchange of credit card numbers over the Internet.
SHA-1 (Secure Hash Algorithm)	the 1994 revision to SHA, developed by NIST, (FIPS 180-1) used with DSS produces a 160-bit hash, similar to MD4, which is very popular and is widely implemented.
Single sign-on	one log-on provides access to all resources of the network.
SKIP (Simple Key for IP)	simple key-management for Internet protocols, developed by Sun Microsystems, Inc.
Skipjack	the 80-bit key encryption algorithm contained in NSA’s Clipper chip.
SKMP (Secure key Management Protocol)	an IBM proposed key-recovery architecture that uses a key encapsulation technique to provide the key and message recovery to a trusted third-party escrow agent.

S/MIME (Secure Multipurpose Mail Extension)	a proposed standard developed by Deming software and RSA Data Security for encrypting and/or authenticating MIME data. S/MIME defines a format for the MIME data, the algorithms that must be used for interoperability (RSA, RC2, SHA-1), and the additional operational concerns such as ANSI X.509 certificates and transport over the Internet.
SNAPI (Secure Network API)	a Netscape driven API for security services that provide ways for resources to be protected against unauthorized users, for communication to be encrypted and authenticated, and for the integrity of information to be verified.
SPKI (Simple Public Key Infrastructure)	an IETF proposed draft standard, (by Ellison, Frantz, and Thomas) public key certificate format, associated signature and other formats, and key acquisition protocol. Recently merged with Ron Rivest's SDSI proposal.
SSH (Secure Shell)	an IETF proposed protocol for securing the transport layer by providing encryption, cryptographic host authentication, and integrity protection.
SSH (Site Security Handbook)	the Working Group (WG) of the Internet Engineering Task Force has been working since 1994 to produce a pair of documents designed to educate the Internet community in the area of security. The first document is a complete reworking of RFC 1244, and is targeted at system and network administrators, as well as decision makers (middle management).
SSL (Secure Socket Layer)	developed by Netscape to provide security and privacy over the Internet. Supports server and client authentication and maintains the security and integrity of the transmission channel. Operates at the transport layer and mimics the "sockets library," allowing it to be application independent. Encrypts the entire communication channel and does not support digital signatures at the message level.
SST (Secure Transaction Technology)	a secure payment protocol developed by Microsoft and Visa as a companion to the PCT protocol.

Stream cipher	a class of symmetric key encryption where transformation can be changed for each symbol of plain text being encrypted, useful for equipment with little memory to buffer data.
STU-III (Secure Telephone Unit)	NSA designed telephone for secure voice and low-speed data communications for use by the U.S. Dept. of Defense and their contractors.
Substitution cipher	the characters of the plain text are substituted with other characters to form the cipher text.
S/WAN (Secure Wide Area Network)	RSA Data Security, Inc. driven specifications for implementing IPSec to ensure interoperability among firewall and TCP/IP products. S/WAN's goal is to use IPSec to allow companies to mix-and-match firewall and TCP/IP stack products to build Internet-based Virtual Private Networks (VPNs).
Symmetric algorithm	a.k.a., conventional, secret key, and single key algorithms; the encryption and decryption key are either the same or can be calculated from one another. Two sub-categories exist - Block and Stream.
TACACS+ (Terminal Access Controller Access Control System)	a protocol that provides remote access authentication, authorization, and related accounting and logging services, used by Cisco Systems.
Timestamping	recording the time of creation or existence of information.
TLS (Transport Layer Security)	an IETF draft, version 1 is based on the Secure Sockets Layer (SSL) version 3.0 protocol, and provides communications privacy over the Internet.
TLSP (Transport Layer Security Protocol)	ISO 10736, draft international standard.
Transposition cipher	the plain text remains the same but the order of the characters is transposed.
Triple DES	an encryption configuration in which the DES algorithm is used three times with three different keys.

Trust	a firm belief or confidence in the honesty, integrity, justice, and/or reliability of a person, company, or other entity.
TTP (Trust Third-Party)	a responsible party in which all participants involved agree upon in advance, to provide a service or function, such as certification, by binding a public key to an entity, time-stamping, or key-escrow.
UEPS (Universal Electronic Payment System)	a smart-card (secure debit card) -based banking application developed for South Africa where poor telephones make on-line verification impossible.
Validation	a means to provide timeliness of authorization to use or manipulate information or resources.
Verification	to authenticate, confirm, or establish accuracy.
VPN (Virtual Private Network)	allows private networks to span from the end-user, across a public network (Internet) directly to the Home Gateway of choice, such as your company's Intranet.
WAKE (Word Auto Key Encryption)	produces a stream of 32-bit words, which can be XORed with plain text stream to produce cipher text, invented by David Wheeler.

Web of Trust	a distributed trust model used by PGP to validate the ownership of a public key where the level of trust is cumulative based on the individual’s knowledge of the “introducers.”
W3C (World Wide Web Consortium)	an international industry consortium founded in 1994 to develop common protocols for the evolution of the World Wide Web.
XOR	exclusive-or operation; a mathematical way to represent differences.
X.509v3	an ITU-T digital certificate that is an internationally recognized electronic document used to prove identity and public key ownership over a communication network. It contains the issuer’s name, the user’s identifying information, and the issuer’s digital signature, as well as other possible extensions in version 3.
X9.17	an ANSI specification that details the methodology for generating random and pseudo-random numbers.

Index