87-20-16 RSA Public Key Cipher, Public Key
Certificates, and E-Mail Privacy Protocols

Alex Bidwell

Payoff

The shortcomings of private key ciphers, and the complexity of establishing secure
protocols for these ciphers, have many systems administrators turning to public key
cryptography to protect the security of confidential data. Systems administrators working
with public key cryptography must learn to balance the goals of users, most of whom
simply wish to protect private information and to verify the sources of electronic mail
messages, with the goals of effective key management, i.e., ease of distribution, reduced
security risk, easy authentication of public keys and key holders, and the easy revocation
of public keys.

Introduction

In public key (i.e., asymmetric) cryptography, a different key is established for the
encryption and decryption of messages. This is designed to render it impossible for the
decryption key for any given message to be inferred from that message's encryption key.
The encryption key, or public key, is placed in the public domain by its owner, so that
anyone wishing to send a private message to the owner can encrypt that message. The
decryption key, or private key, is guarded to protect against any hostile decryption
attempts. This article discusses the merits of a standard public key cipher, Rivest Shamir-
Adleman, and its implications in terms of electronic mail security.

RSA Public Key Ciphers

Although many public key ciphers have been proposed, only a few are both secure and
practical. Rivest Shamir-Adleman, in particular, works well for both encryption and digital
signatures, although its encryption and decryption speeds are slower, by a factor of 100,
than those of a private key cipher such as the Data Encryption Standard (DES).

The security of RSA is based on the difficulty of factoring very large numbers. Take,
for example, two large prime numbers, p and q, that are each several hundred bits long.
Their product is n = pq. Choose a random integer, e, relatively prime to (p–1)(q–1) and d
such that ed = 1(mod(p–1)(q–1)).

The pair (e,n) is used as the public key and may be distributed or published as its
owner sees fit; the pair (d,n) is used as the private key and d is kept secret. Furthermore,
the prime numbers, p and q, must be kept secret, or be securely discarded, after n is
generated.

Encryption of plaintext, or P, using the public key, (e,n), yields ciphertext, or C:C  Pe
(mod n).

Decryption of the ciphertext, C, with the private key, (d,n), recovers the plaintext using
the same operation except with d as the exponent: P  Cd (mod n) = (Pd (mod n))=(Ped
(mod n))=P (mod n).

RSA has two significant advantages. The first is that it makes key distribution simple.
Each key owner publishes a public encryption key in a public directory for use by those
wishing to send them an encrypted message, or sends the encryption key to these senders
directly. The second advantage is that the private decryption key is held only by the owner
and is therefore less vulnerable to theft than a shared key. Again, the main disadvantage of
RSA is that encryption an decryption are more time-consuming than with private key
algorithms.
One method used by systems administrators to get around the performance problems of public key cryptography, and the key distribution problem associated with private key cryptography, is by using both public and private keys simultaneously. This idea can be illustrated as follows: If A wants to send a private message to B, A generates a random private, or symmetric, key, K, and encrypts the message using a private key cipher, such as Data Encryption Standard. A then encrypts the private, or symmetric, key used with B’s public key, and forms the following ciphertext string to send to B: EB{K}, K{M}; ciphertext is sent to B.

B may then use his private key, DB, to recover K. B knows it will be the first 56 bits. DB{EB{K}} = K; K is recovered.

B may then use K to decrypt the remainder of the message. K-1{K{M}} = M; M is recovered.

**Digital Signatures, RSA, and Digital Signature Algorithm (DSA)**

Digital signatures can be implemented with public key cryptography to authenticate the source of a message and its integrity. If the encryption and decryption operations are commutative, the order in which they are performed should yield the same result: DX(EX(M)) = EX(DX(M)) = M.

This idea has some interesting implications. If A processes a message, M, with his private key, DX, it creates a unique image of M based on A’s private key, i.e., a signature. DX(M) = SX(signature).

Then, if the message and the signature (M, SX) are sent to B, B can calculate to recover the plaintext. EX(SX) = EX(DX(M)) = M (message, or recovered plaintext).

Recovery of the plaintext, message M, sent with the signature S, proves that A signed the original message, M, because only A has knowledge of the private key, DX. This verifies the message’s integrity and authenticity. However, the amount of time it takes to perform the signing and authentication operations with RSA public key cipher, in instances where messages are lengthy, has caused some concern.

With a public key cipher such as RSA, the entire message, M, and signature, S, must be processed to recover the plaintext, which then must be compared to the original message, M. This task could be performed much faster if the public key cipher could operate on a short representation, or digest, of the message, M. This is an especially effective solution if it is not necessary to keep the message secret, but only to verify its integrity and authenticity.

To implement this solution, the National Institute of Standards and Technology has proposed a digital signature algorithm (DSA) as a standard for authentication only. Like Rivest-Shamir-Adleman, DSA uses a public key approach. However, because it is not usable for encryption, the US government allows it to be exported; RSA, conversely, cannot be exported.

**Public Key Certificates**

When a user sends a private E-mail message using the public key cryptography systems available today—for example, privacy enhanced mail (PEM) or Pretty Good Privacy (PGP)—they must obtain the recipient’s public key. For recipients to verify the sender’s electronic signature on a document, they must have the sender’s public key. To obtain someone’s public key, yet be assured that no one has tampered with it (e.g., that a phony key has not been issued to impersonate the sender), systems administrators must accept a public key only if it is certified. Certifying authorities, or trusted third parties, create a certificate by notarizing the key holder’s public key with their signature after validating the
key holder's proof of identity. The certificate is a bond between public key holders and
their public key that is vouched for by the certifier.

The validity of public key certificates must be checked before the public keys are used.
If the certifier's signature does not check out, the public key or other parts of the certificate
(e.g., the name) have likely been altered. The manner in which trust is conferred to the
certifying authority is called a trust model.

**Privacy Enhanced Mail (PEM) and Pretty Good Privacy (PGP)**

Two secure E-mail systems are currently being implemented: Privacy Enhanced Mail
(PEM) and Pretty Good Privacy (PGP). Each one uses a different trust model to validate
certifying authorities. PEM, for example, has a trust model based on a hierarchical structure
of certifying authorities.

**Privacy Enhanced Mail (PEM)**

PEM is the standard proposed by the Internet Engineering Task Force (IETF) that
defines procedures for message encryption and authentication services (via digital
signatures) for electronic mail transfer on the Internet. PEM is specified by Internet

- RFC1422—Part 2: Certificate Based Key Management.
- RFC1424—Part 4: Key Certification and Related Services.

PEM is compliant with the Public Key Cryptography Standards (PKCS) developed by a
consortium headed by Rivest_Shamir-Adleman Data Security, Inc. and several software
developers, including Apple, Novell, Lotus, Microsoft, Fischer International, and Sun
Micro-systems.

PEM is capable of specifying the cipher algorithms it is using. It puts all messages into
a canonical form before any cipher or hash operations are performed on them, and ensures
that secure E-mail can be interchanged following PEM Internet standards. PEM enables
users to do the following:

- Ensure the privacy of their E-mail using a private(symmetric) key cipher algorithm,
such as Data Encryption Standard.
- Securely distribute symmetric keys using RSA for key encryption.
- Validate public key certificates that are digitally signed using RSA.
- Check message integrity using a hashing algorithm (MD2 or MD5).
- Check digital signatures for authentication or nonrepudiation using RSA on a MD2 or
  MD5 hash of the message.

PEM messages utilize unencrypted headers to identify what type of processing was
performed and which algorithms were used. The fields in the headers contain more
information for the recipient to use when determining the validity of the message, including
the public key certificate, the encrypted symmetric key for encoded messages, the message integrity check (MIC) field to indicate the validity of a message and whether it has been digitally signed, and the message itself. The data encryption key, MIC, and the message are encrypted only if indicated in these header fields.

**Issuing PEM Public Key Certificates**

A certifying authority (CA) can be any trusted central administrator willing to vouch for the identities of those whose keys it certifies. CAs should follow the comprehensive Consultive Committee on International Telegraphy and Telephony (CCITT) X.509 standard recommended for certificates and CAs. The CA may be the only party from which to obtain a public-private key pair in certain high-security locations. In a company setting, a potential employee's information is verified at the time of employment, so usually only employees' and their managers' signatures are required on a form. If an employee can generate a public-private key pair, he or she sends the self-signed public key with required proof of identity to an appropriate company CA to be certified. CAs may issue certificates based on varying levels of identification verification of the key holder.

Online CAs are becoming increasingly necessary to satisfy the growing security demands of the information superhighway. Banks and major credit card companies will likely assume this role and issue their own public-private key pairs. These companies may well use identity verification as their sole key certification. Exhibit 1 presents an example of a PEM hierarchy of certification authority.

**PEM Certification Authority Hierarchy**

**Determining When PEM Public Keys Are No Longer Valid**

PEM keys are deemed invalid in the following instances:

- Before the start date and after the expiration date indicated on their certificates. The dates in a certificate must be checked with the date of the transaction.

- When they have been revoked by the CA and placed on a public list known as the certificate revocation list (CRL). Revocation is detectable only if the party using the public key checks the CRL. The CA should provide this facility or indicate where and how the CRL can be checked.

- When they have been revoked by the key holder or key issuer and placed on a CRL. As previously mentioned, revocation is detectable only if the party needing to use the public key checks the proper CRL. Notably, invalid keys can be used to perform all the operations discussed here. However, a properly designed application should check for and require valid keys.

**Revoking PEM Public Keys**

Public keys are revoked for the following reasons:

- To prevent misuse, including forgery, if the security of the key holder's private key has been compromised.

- To prevent misuse, including man-in-the-middle attacks, if the security of the public key server has been compromised.
To prevent validation of a signature of an employee who separates from an organization. This does not, however, prevent the ex-employee from signing documents and including the no-longer-valid certificate.

To prevent key holders from authenticating themselves to servers that check CRLs.

RSA has had a certificate services center in operation since 1993 which certifies CAs, issues certificates, and provides CRL services. BBN Communications manufactures The SafeKeeper Certificate Issuing System, and Trusted Information Systems provides certification services and products to support certifying authorities and maintains a CRL on an E-mail accessible server. The future will likely bring central or regional repositories of CRLs (e.g., banks or credit card companies). These trusted third parties can be used to run an exhaustive check on whether a public key's certification has been revoked. The US Postal Service is interested in acting as a policy CA and providing electronic postal services which would include certifying public keys, sealing electronic documents with digital signatures, and encryption services.

Checking Public Key Certificates and Certificate Revocation Lists

There are several situations in which certificates and CRLs should be checked without hesitation.

- When the message, message source, or subject of the transaction warrants further verification before it is acted on. PEM protocol indicates the certificate and CRLs should be checked before using any party's public key to encrypt a message to be sent to that party.

- If unauthorized server access could jeopardize the security of other network nodes, such as a bank serving a merchant's server.

- When security policy demands it.

With PEM, the capability to check certificate validity easily is important and is part of the protocol if properly implemented. The rapid accessibility of CRLs is essential for PEM to be useful and trusted. It is important to be able to understand the level of verification used by CAs when they have checked a user's proof of identification and key validity. Not all CAs use the same identity certification criteria. The current certificate structures do not provide for such an indication.

The need to check certificates and CRLs raises several questions that need to be addressed in all PEM certification security strategies, including:

- How and where can a CA's certification policy or its CA's policy be checked?

- Are there any standards or ratings for levels of confidence in a CA's identification verification levels?

- How safe and secure are a CA's private certifying keys?

- Are the CRLs available online without great delay?

- How often are the CRLs updated?
Will old CRLs be available for checking or audit?

**Pretty Good Privacy (PGP)**

The latest versions of PGP provide the same security functions provided by PEM—confidentiality, integrity, signatory authentication, and nonrepudiation. PGP utilizes the International Data Encryption Algorithm (IDEA), instead of DES, for message encryption and MD5 for hashing. RSA is utilized for operations requiring a public key algorithm. Public key certification is based on the creation and maintenance of each individual user's web of trust. Distribution of private (symmetric) keys is similar to PEM; they are encrypted using the public key of the person (or process) to whom they are to be distributed. Message encapsulation is different from encapsulation in PEM. PGP uses self-defining headers, which can be nested and, therefore, encrypted. Hence, less information is available for a traffic analysis attack. PGP uses a different trust model and certificate structure than PEM. PGP's trust model is a personal referential web structure that is uniquely established by every user. Every user's web is different. Because this model does not require the existence of a formal CA infrastructure, PGP has experienced faster growth than PEM.

**PGP Public Key Certificates**

As with PEM, PGP certificates provide an association between a public key and its owner whose association and key validity are attested to by those users who have digitally signed the public key. Pretty Good Privacy public key pairs are generated by individual users.

The PGP public key certificate consists of two parts: the public key and the owner's user ID and address (bidwell@prodigy.com), and the digital signatures of the key owner and other users who need the owner's key. The key owner should sign the key to prevent forgery, or anyone could generate a key and associate them with the owner's user ID and address.

**Issuing PGP Public Key Certificates**

PGP public key certificates are issued by other users. With PGP, each user is a CA. If users need to check someone's digital signature on a document, they will want to use that person's public key only if it has been certified by someone they trust. The users have several options. They can contact others they trust to see whether this trusted third party has certified the public key, they can contact a PGP public key server, or they can contact the actual person. Once the public key is certified, PGP adds the new key to a user's keyring with its signature and parameters, which indicates their belief in the validity of the key. The key holder is then established as a trusted third party and is permitted to introduce other key holders. When the new key is added, a new link is added to the user's trust graph. This ever-widening graph creates the web of trust.

When a user receives the public key of another party, PGP checks to see whether any of the certifiers of the key are included on the user's keyring. If any certifiers are on the user's keyring, PGP checks the level of trust that those certifiers have been given to act as introducers of other key holders. The established levels of trust are used by PGP to determine the validity of the public key under question.

Key validity can also be established by personal knowledge of the key value, user ID, or by checking the PGP fingerprint of the key. The fingerprint is simply the 128-bit binary value (or 16 hexadecimal characters) of the MD5 hash value of the public key. If this can be established by some trusted means and successfully compared with the calculated hash value, validity is established.
To obtain keys that have been posted, users can access PGP public key servers via the Internet using the World Wide Web, File Transfer Protocol, or E-mail. Many keys include a fingerprint. If the transaction does not require absolute security, the validity of the fingerprint can be accepted and checked against the calculated value to determine key validity.

**Determining When PGP Keys Are No Longer Valid**

Keys are valid until they are revoked by their owner, who issues a key revocation certificate. However, if a user receives a key revocation certificate from, for example, Joe, the user should personally (not inband via E-mail) verify it with Joe before incorporating it onto his keyring, because key revocation is irreversible. Joe's old key will never be usable from the user's keyring after it is revoked.

**Revoking PGP Public Keys**

The same reasons presented for PEM revocation apply to PGP. If the private key of a public-private key pair is compromised, users must take steps to ensure that their communications will not be further compromised. Any messages that are sent to the user and encrypted with the user's public (encryption) key can now be decrypted by the culprits who stole the user's private key if they can get a copy of the E-mail message. Even worse, the culprits can digitally sign documents with the user's signature.

Users must notify anyone who might send confidential communications if their key has been revoked. They do this by distributing to likely correspondents a key revocation certificate and a new public key signed by the user to prevent key forgery. If users use PGP public key servers where they have posted their public key, the problem should be solved. However, some people do not check their E-mail, have passed the user's key to their friends, or use key servers that are not well run and do not post key revocations promptly. Current versions of PGP check the user's local keyring for key revocation certificates, but they do not check elsewhere (e.g., a public key server or a CA's CRL, as prescribed in PEM).

There are several concerns surrounding PGP’s operations in this area.

- How rapidly do PGP public key servers post key revocation certificates?
- Will PGP public key servers be able to maintain readily accessible CRLs on the Internet?
- Will PGP- and PEM-generated keys become interoperable?
- Are PGP public key servers secure enough to prevent key substitution, which would enable a man-in-the-middle attack?

**Conclusion**

Protocols used on the Internet are becoming increasingly more sophisticated. Secure protocol http (s-http protocol) used in many of the net browsers available today provides for negotiation of algorithms, modes, and parameters, including:

- Encapsulation format: PEM, Pretty Good Privacy, or PKCS-7
- Signature algorithm: Rivest_Shamir-Adleman or DSA
· Key exchange: RSA, Inband, D-H, Kerberos, Outband
· Message digest algorithm: MD2, MD5, SHA
· Encryption algorithms: Data Encryption Standard, Interactive Data Extraction and Analysis, RC2, RC4
· Protection mode: Signature, Encryption, Keyed MAC
· Public key certificate format: X.509 or PKCS-6
· Certification pattern: MasterCard or VISA

Not all transactions require the same level of verification. This concept is important for the Internet, where both ease of use and security of transactions must be possible. There will always be varying levels of security required, because many users transmit noncritical information.

Today, most functions on the Internet operate on the first level above having little or no security requirements. As soon as users get beyond that level, they see that the most important issues involving key management remain distribution, revocation, and validity checking. These issues must be successfully resolved to achieve safe electronic commerce.

Author Biographies
Alex Bidwell
Alex Bidwell is president of CyberQuest, Inc.