RIPEM Beginner's Guide

Gary Ratterree
Texas A&M University

garyr@cs.tamu.edu

March 1993

Introduction to Cryptography

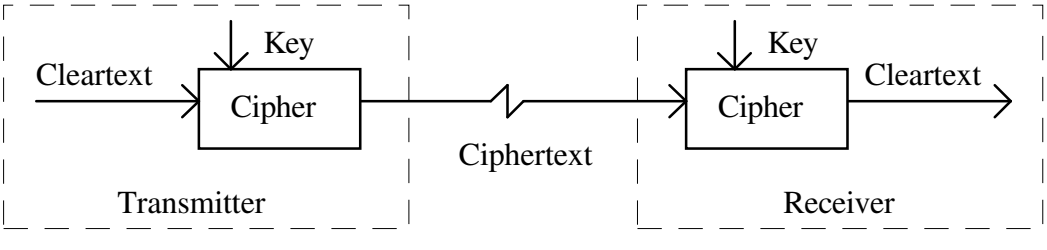
An understanding of the basic principals of cryptography is necessary in order to understand how to use the RIPEM mail security program. This Beginner's Guide gives a short introduction to the topic. It also describes how to install RIPEM on UNIX systems. Users are encouraged to consult the RIPEM User's Guide and the references in the bibliography for more information on RIPEM and encryption systems.

Conventional Cryptography

Conventional, or symmetric, cryptography is what most people think of when they think of "secret codes". In conventional cryptography, a single key is used to encrypt (encode) a message. The same key is then used to decrypt (decode) the message. Messages that are going to be encrypted are traditionally called cleartext. The message is called ciphertext after it has been encrypted. The mathematical algorithm that performs the encryption is called a cipher. An example of a commonly used conventional cipher is the U.S. Data Encryption Standard algorithm (DES).

Conventional ciphers have the advantage of being very fast. Large messages can be encrypted using conventional ciphers in seconds with modern computer systems. The disadvantage of this system is that both parties in the conversation must know what the key is. There must be some secure method of sending the key from the sender to the receiver. If someone ever discovers what the key is, they can decrypt any message that is being sent. RIPEM uses the DES algorithm to protect the contents of a mail message from being decoded by anyone other than its intended recipient.

Figure 1. A Symmetric or Conventional Cipher.



(Adapted from Voydock and Kent p. 15)

Public Key Cryptography

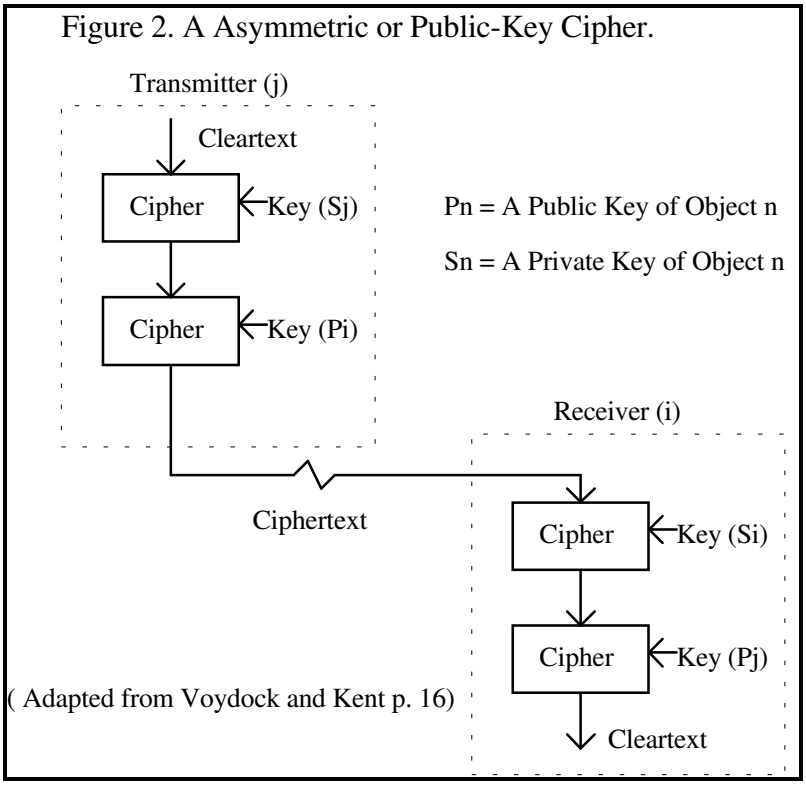
Public-key, or asymmetric, cryptography uses two keys to encrypt and decrypt messages. One of the keys is called the private key and the other is called the public key. Messages that are encrypted with the public key can only be decrypted with the private key. Messages that are encrypted with the private key can only be decrypted with the public key. This dual key feature of public key cryptography makes it useful for message exchanging. The best known public-key cipher is the RSA algorithm.

If one of the keys is kept secret, the other key can be given to anyone without fear of compromising the system. The public key is given away to all the people that you wish to communicate with. In the RIPEM system public keys are registered with an Internet database server. This server then provides automatic key lookups. This is the major advantage of public-key cryptography. Public keys are not secret, so key distribution is much easier.

When we want to send a message to another user, we can encrypt it with their public key. This insures that only the intended recipient can read the message with their secret key. Most public key systems take this concept one step further. We encrypt the message with our secret key first and then we encrypt with the recipients public key. Figure 2 illustrates this procedure. The recipient first uses his secret key to decrypt the message. The receiver then must also use the sender's public key to decrypt the message. This assures the receiver that the sender actually sent the message, since only the sender knows his secret key. This dual encryption method insures both secrecy and authentication.

The major disadvantage of public-key encryption is that it is very slow. For this reason, messages are often encoded with a conventional cipher. A public key cipher is then used to send the conventional key and a checksum. The key to the conventional cipher is encrypted with the receiver's public key. Only the receiver has the private key that can decipher the message. The checksum is then encrypted the sender's private key. The checksum insures that the message has not be tampered with, and it insures the receiver that the sender actually sent the message. The relative ease of key exchange is the method that RIPEM uses to send messages. RIPEM's encryption scheme will be discussed in greater detail in the next section.

Figure 2. A Asymmetric or Public-Key Cipher.



How RIPEM uses Cryptography

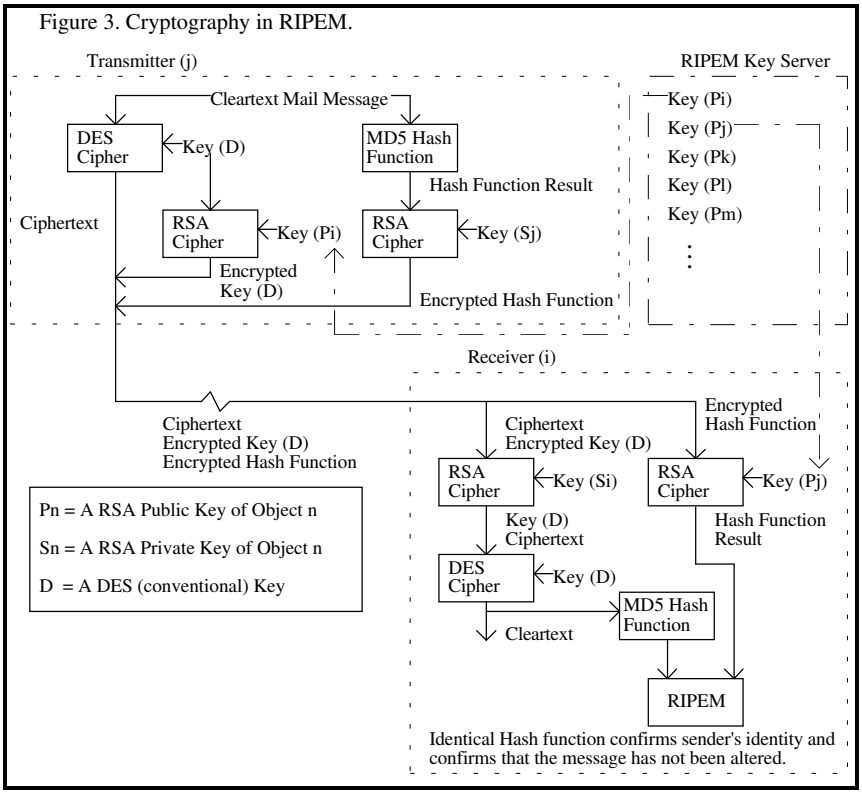
RIPEM uses a combination of both conventional and public key encryption systems to send secure electronic mail messages. In order for the RIPEM system to be valid, it must insure that only the intended recipient of a message can read the message. It must also protect against willful changes to the message and transmission errors. It also should insure that the message comes from the person that it says it does.

The message itself is encrypted with the U.S. Data Encryption Standard cipher (DES). This is a conventional cipher that uses a 56 bit key. The DES algorithm is used to encrypt the message because it is much faster than a public key encryption system. The DES key is generated randomly by the RIPEM program. No two identical DES keys are ever generated. The DES key is then encrypted by RIPEM using the RSA public key encryption system. The sending RIPEM program uses the recipient's public key to encrypt the DES key. RIPEM obtains this key from the RIPEM key server program. Only the intended recipient will be able to decipher the DES key with their secret key. If there is more than one recipient, the DES key must be encrypted with each recipient's public key.

A cryptographically secure hash function is also used to protect the message. The MD5 algorithm is used to create a hash value, or checksum, of the plaintext message. This hash function will detect any changes made to the original text. The sending RIPEM program encrypts this hash value with the sender's private key. The DES encrypted plaintext, the RSA encrypted DES key, and the RSA encrypted MD5 hash result is then sent to the recipient via normal electronic mail.

When the recipient receives a RIPEM message, they first must use their secret key to decrypt the DES encryption key. The DES key is then used to decipher the message. RIPEM then computes the MD5 hash value for this message. It also deciphers the MD5 hash function result with the sender's public key which it obtains from the key server. If the two hash functions are equal, then the message is guaranteed not to have been tampered with during transmission. This is true because the MD5 has function will catch any modification to the message. The message is also guaranteed to be from the sender. This is true because only the sender could have encrypted the correct MD5 value with the correct secret key. Figure 3 illustrates how RIPEM sends a message.

Figure 3. Cryptography in RIPEM.



The obvious weakness with RIPEM is in the distribution of public keys. If a agent can substitute their public key in place of yours, then they can read all messages addressed to you. The current RIPEM key server is fairly weak in this regard. You should periodically check your public key on the server to make sure that it has not been tampered with. There are also several other potential security problems in the RIPEM system. Users are advised to read the RIPEM User's Manual and the RIPEM FAQ on Attacks. Both are included in the standard RIPEM distribution. The latest FAQ and other RIPEM information is also available on the Usenet news group alt.security.ripem. These documents describe possible attacks against RIPEM in great detail and they present ways to protect against them..

Using RIPEM on the Texas A&M Computer Science Systems

This section suggests techniques for using RIPEM in conjunction with popular UNIX mail programs. Use of the C-Shell is assumed.

It is possible, of course, to compose a message in a text editor, save the message to a file, run RIPEM to encrypt the message, start your mailer, insert the encrypted file into a message, and then send the message. In fact, the encryption and mailing can be done on separate systems, with appropriate file transfers. However, on most UNIX systems it is possible to eliminate several of these tedious steps.

Setting Up Your Environment

It is recommended that Internet-connected UNIX users include the following in their `.cshrc` file (or on the Texas A&M Computer Science systems place in the .login.sun4 file):

```
setenv RIPEM_SERVER_NAME      dsls4.cs.tamu.edu
setenv RIPEM_PRIVATE_KEY_FILE ~/.ripemprv
setenv RIPEM_PUBLIC_KEY_FILE  ~/.ripempub
setenv RIPEM_USER_NAME        (Your email address; e.g., garyr@cs.tamu.edu)
```

You can set up multiple e-mail address for RIPEM as follows:
`setenv RIPEM_USER_NAME garyr@cs.tamu.edu,garyr@cs,garyr,std_grr@shsu.edu`
The first entry should be your primary Internet address.

Create a shell script to encrypt RIPEM messages. Place the following lines in a file named `ripem-encrypt`, put this file in a directory mentioned in your path, and **give it execute permission** (`chmod 700 ripem-encrypt`). (This file is available in the RIPEM distribution, in the `util` directory.)

```
#!/bin/sh
tempfile=/tmp/msg-`whoami`
ripem -e -h pr -i $1 -o $tempfile
cp $tempfile $1
rm $tempfile
```

Create a shell script to decrypt RIPEM messages. As above, place these lines in the file `ripemd`:

```
ripem -d | more
```

Create a shell script to help reply to encrypted RIPEM messages. Place this in `ripemr`:

```
ripem -d -h pr | quote -h
```

Include the following lines in the file `.mailrc` in your home directory. (A sample `.mailrc` can be found in the RIPEM distribution, in the `util` directory.)

```
set editheaders
set EDITOR=ripem-encrypt
```

Copy the correct RIPEM executable for the machine that you wish to use from the directory /pub/advos/RIPEM/executables. There are compiled versions here for several platforms. The executable should be in a directory referred to in your path statement. Version 1.0.5 is the latest version of RIPEM, other versions are provided for compatibility reasons.

Creating a RIPEM Public Key

The initial generation of keys can proceed something like this:

```
cd ~
ripem -g -S .ripemprv -P .ripempub -R eks
```

Type random garbage at the keyboard when prompted.

Type in a secret password when prompted for the password to your private key. Type it again when prompted for verification. This password can contain up to 255 characters and may contain spaces.

If you set up RIPEM for different e-mail addresses, you should modify BOTH `.ripempub` and `.ripemprv` to look like the following (your primary user name must be the first entry):

```
-----BEGIN PUBLIC KEY-----
User: garyr@cs.tamu.edu
User: garyr@cs
User: garyr
PublicKeyInfo:
 MFgwCgYEVQgBAQICAf8DSgAwRwJAbiMDTw/j/q7H7hUDBFYV/+8NBBJM/IsaIY1G
 s6SByD1NRBqCkblUq4iYETsW0/JQKieZYwYntNAgu3+bRIuUxwIDAQAB
MD5OfPublicKey: 5A677077AB7C6D9DD358224A72735CC7
-----END PUBLIC KEY-----
```

Now register the key with the Texas A&M Internet key server:

```
mail ripem-register-keys@cs.tamu.edu <.ripempub
```

You should not register public keys that have simple e-mail addresses with remote key servers. For example, I should only register `User: garyr@cs.tamu.edu` with the key server ripem.msu.edu. Garyr and garyr@cs should not be registered on remote servers since these addresses could apply to many other users on the Internet.

An alternative key server that you can use is ripem.msu.edu. Please see the RIPEM User's Guide for information on changing and deleting your key.

Encrypting an Email Message Using "mail"

If you are using the "`mail`" package that comes with many UNIX systems, you can use the following procedure to compose, encrypt, and send a message. In this example, your input is in bold.

```
$ mail smith@bigu.edu
Subject: Greetings
This is a test message
~e
Enter password to private key: (Type your password here.)
(Type Control-D)
$
```

The ~e command to mail was originally designed to **e**dit the message being composed. Nowadays, however, it is rarely used (in favor of the ~v **v**isual edit command). The tricks described above effectively turn ~e into an "**e**ncrypt" command on SunOS and some other UNIX systems. On some UNIX systems, however, the mail command does not interpret the editheaders option. On those systems, you need to use a different approach, which requires you to type the recipient's address twice:

```
$ mail smith@bigu.edu
Subject: Greetings
This is a test message
~| ripem -e -r smith@bigu.edu
Enter password to private key: (Type your password here.)
(continue)
(Type Control-D)
$
```

Decrypting an Email Message Using "mail"

```
$ mail
Mail version ......
"/usr/spool/mail/jones": 1 message 1 new
>N  1 smith@bigu.edu       Wed Sep 30 22:38    29/1119  Greetings
& pipe ripemd
Pipe to: "ripemd"
Enter password to private key: (Type your password here.)
(The plaintext message is displayed.)
"ripemd" 29/1119
& q
$
```

Changing Your Private Key Password

You can change the password to your private key by executing the following commands on an UNIX system:

```
ripem  -c  -s .ripemprv  -S newprv
      Enter password to private key:       (Type your old password here.)
      Enter new password to private key:   (Type your new password here.)
      Enter again to verify:               (Type your new password again to verify.)

mv  newprv  .ripemprv
```

This changes only the password to the key, not the key itself. If you feel someone has gained access to the key, then you need to generate a new key. Changing the password should be used if your password has been compromised, but the private key is still secure.

Changing Your Public Key on the Key Server

To change your public key, you must first create a new key pair with RIPEM and then sign the new public key with your old private key. You then send the new key to the server which will automatically update your key.

Follow these steps on an UNIX system:

```
ripem -g -P newpub -S newprv -R eks
```
(*This will create two new key files.*)

```
ripem -e -m mic-only -i newpub -o signedpub
```
(*This signs the new key with your old key.*)

```
mail ripem-change-keys@cs.tamu.edu<signedpub
```
(*This mails the new key to the RIPEM key server.*)

You should wait until the key server sends you back a message indicating that your key has been successfully registered before deleting your old keys. You may wish to keep your old secret key around to decrypt any messages that might have been sent before you changed your key.

Follow these steps on an UNIX system:

```
mv .ripemprv oldprvkey

mv newpub .ripempub

mv newprv .ripemprv

rm signedpub
```

Removing Your Public Key from the Key Server

To delete your public key from the RIPEM key server you should create an ASCII file named remove.key and place the command `RemoveKey` in it. You then need to sign this file with the RIPEM program. The following command should do this:

```
ripem -e -m mic-only -i remove.key -o remove.sig
```

You then need to mail the file remove.sig to the key server. The following command should do this on UNIX systems:

```
mail ripem-remove-keys@cs.tamu.edu < remove.sig
```

If the key was successfully removed, the key server will send you back a message indicating that the key was removed successfully.

Getting a List of All Public Keys on the Server

Sending an e-mail message to `ripem-get-keys@cs.tamu.edu` will return to you a list of all public keys currently registered on the key server. You can use this to find the users that you can send RIPEM messages to. It is also useful for checking your public key. The MD5 value that is returned by the key server for your public key should match the one in your local public key file (`.ripempub`). If they do not, then it is possible that your public key has been changed in an attempt to intercept your messages. You should delete your current key using the procedure described above and create a new key pair.

BIBLIOGRAPHY

Blair, Charles. *Notes on Cryptography.* Business Administration Dept., Univ. of Illinois, 1991.

Fabin, Paul. *Answers to Frequently Asked Questions About Today's Cryptography.* RSA Laboratories, September 1992.

Riordan, Mark. *RIPEM User's Guide.* RIPEM 1.0.5 Distribution, December, 1992.

Riordan, Mark. *RIPEM Key Server.* RIPEM 1.0.5 Distribution, September, 1992.

Voydock, Victor, and Kent, Stephen T. "Security in High-Level Network Protocols." *IEEE Communications Magazine*, Vol. 23, no. 7, p. 12-23, July 1985.

Woo, Thomas, and Lam, Simon. "Authentication for Distributed Systems." *IEEE Computer*, Vol. 25, no. 1, p. 39-52, January 1992.