

# Table of Contents

<b>Part I Introduction</b>	<b>3</b>
1 What is SecExMail Gate? .....	3
2 Overview .....	5
<b>Part II Configuration</b>	<b>8</b>
1 Conventional SMTP clients - LAN .....	8
2 SecExMail SMTP clients - Internet .....	10
3 Conventional POP3 clients - LAN .....	11
4 SecExMail POP3 clients - Internet .....	12
5 Secure Socket Layer & Certificates .....	13
6 Entropy Collection Settings .....	14
7 System Service Settings .....	16
8 Personal and Department Keys .....	17
9 Group Key Management .....	19
10 Friend Keys Management .....	20
11 System Event Log .....	22
12 Client Event Logs .....	22
<b>Part III Keys</b>	<b>24</b>
1 Create your personal SecExMail keys .....	24
2 Personal Details Screen .....	25
3 Key Size Screen .....	25
4 Passphrase Screen .....	26
5 Entropy Screen .....	27
6 Progress Screen .....	27
<b>Part IV Certificates</b>	<b>28</b>
1 Generating Certificate Requests .....	28
2 Installing Certificates .....	33
<b>Part V Technical</b>	<b>35</b>
1 RSA Public Key Encryption .....	35
2 SecExMail Encryption .....	35
3 ISAAC Random Number Generator .....	38
4 SecExMail Message Format .....	38
5 SecExMail Keys .....	40

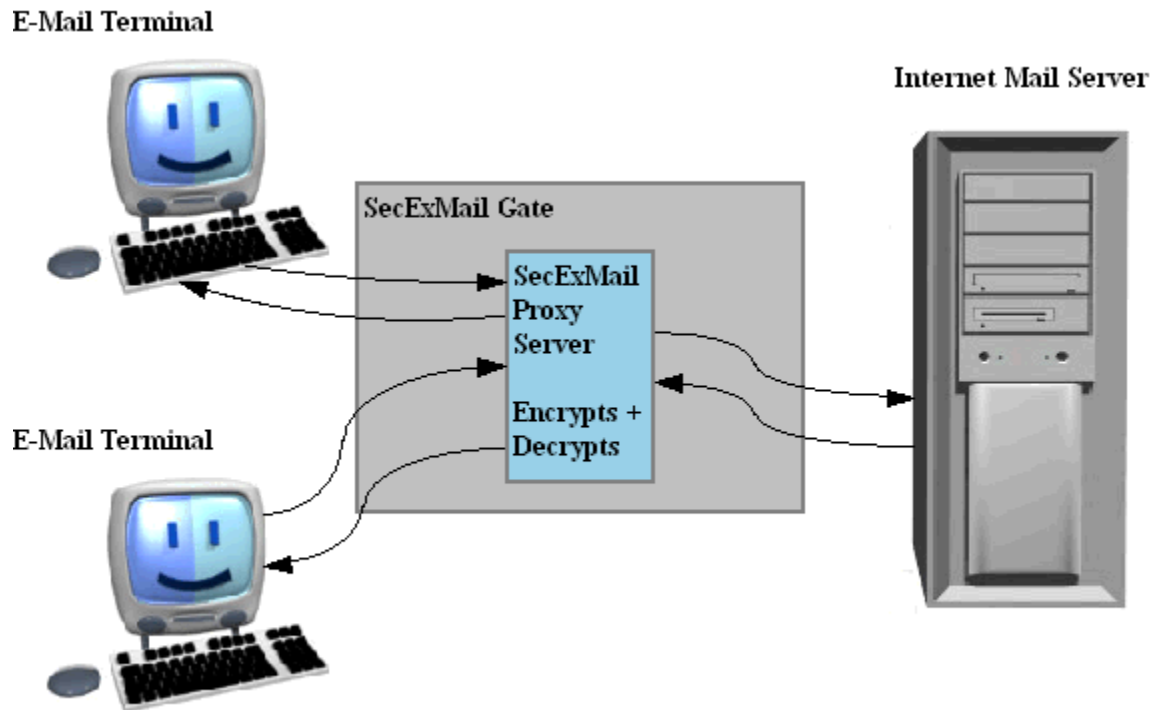
6	SecExMail Key File Format .....	40
7	Entropy Collection .....	42
8	One-Time Pads .....	46
9	IP / DNS Spoofing .....	46
10	Requirements .....	47
11	Known Plain Text Attack .....	48
<b>Part VI About</b>		<b>49</b>
1	About SecExMail Gate .....	49
2	About Bytefusion Ltd. ....	49
3	Acknowledgements .....	50
	<b>Index</b>	<b>0</b>

# 1 Introduction

## 1.1 What is SecExMail Gate?

SecExMail Gate is a *server based* e-mail proxy server which encrypts and decrypts email messages on your corporate network in real-time. This proxy server integrates with your existing corporate infrastructure and operates in conjunction with your SMTP/POP3 mail server. SecExMail Gate implements open standard encryption algorithms to create a seamless security framework to protect the privacy of your corporate email on the public internet. No additional software on the e-mail workstations on your network is required.

### SecExMail Network Diagram



### General Features :

- Easy to configure

SecExMail Gate requires no plug-ins or other email client specific software. Simply configure SecExMail Gate to communicate with your email server and set your favorite email client to talk to SecExMail Gate. That's it.

- Seamless integration

Probably the greatest obstacle to wide-spread use of secure email is that most encryption systems don't integrate seamlessly with popular email clients. In some cases the use of plug-ins means that

encrypted messages held in mail folders are not searchable via the standard email client interface. In other cases, encrypted messages held in mail folders become irretrievable once the encryption plug-in is unloaded. In most cases security is an after-thought and the normal work flow is disrupted to accommodate security. Because SecExMail Gate operates unobtrusively in the background, encrypting and decrypting email streams to and from your email client in real-time, you continue to work with your email client as usual.

- Fail Safety

Many plug-in based encryption systems require the user to treat secure mail differently from ordinary mail. Some require the user to remember that mail to a specific recipient should always be encrypted and take special action to invoke the encryption. If the user forgets, the message is sent in plain text and confidential information may be compromised. Equally, if a plug-in is accidentally unloaded or crashes, sensitive information may be compromised because messages designated secure are inadvertently spilled onto the internet in clear text. SecExMail Gate is engineered from the ground up to provide fail safety. Because SecExMail Gate acts as a relay agent or mail proxy and requires the email client to be configured to communicate with its mail server via this proxy, a failure in SecExMail Gate simply means that no mail is sent until the error condition is alleviated. Once the public key for a particular recipient is entered into SecExMail Gate, all mail to that recipient will be sent encrypted by default and without further user intervention.

- Proactive Security

SecExMail Gate does not stop at simply encrypting your email messages. It also provides for message stealth at the protocol level. The information contained in the header of most emails provides a wealth of information to the cryptanalyst. For example, the header contains a subject line which tells the cryptanalyst which messages are worth examining. Furthermore the header contains information about the type of message being sent, the so called "MIME type". The MIME type indicates to the cryptanalyst if the message contains only text or perhaps a photograph and if so in what format the photograph is stored ( JPG, GIF, etc ). The latter can be exploited in a [known plain text attack](#)<sup>[48]</sup>. For this reason, SecExMail Gate not only encrypts the message subject but also obscures MIME type information. This means a hacker can neither deduce whether the message is worth examining nor what file attachments, if any, are being sent.

- Protect Account Information

Most conventional email communication involves the exchange of clear text passwords. This means that anyone with the right wire tapping equipment, or in fact any skilled system administrator working for your telecommunications company, can collect your password information and subsequently read all your email without your knowledge. SecExMail Gate can protect your user and password information by encapsulating all email traffic in a Secure Socket Layer ( SSL ) or Transport Layer Security (TLS) tunnel.

- Trojan Horse Protection

SecExMail Gate protects against attempts to trick you into revealing your password information to third parties. See IP/DNS spoofing for technical details. (Offshore and Corporate edition only )

- Key Transparency

SecExMail Gate is engineered with a focus on transparency to give you the assurance that no backdoor keys or key recovery is embedded in encrypted messages.

### **Technical Features :**

- Public Key Encryption

SecExMail Gate uses standard [RSA based public key encryption](#)<sup>[35]</sup>. Supported key sizes are 2048, 4096 and 8192 bits ( up to 10240 bits for offshore edition and corporate edition ). Two messages are never encrypted with the same session key. Instead the public key associated with the recipient of a message is used to encrypt a random session key which is used to encrypt the message. Generation of strong session keys is based on a sophisticated [entropy collection system](#)<sup>[42]</sup>.

- Message Encryption

Individual messages are double encrypted via 64 bit ISAAC and 256 bit Twofish encryption .  
[See SecExMail Encryption](#)<sup>[35]</sup>.

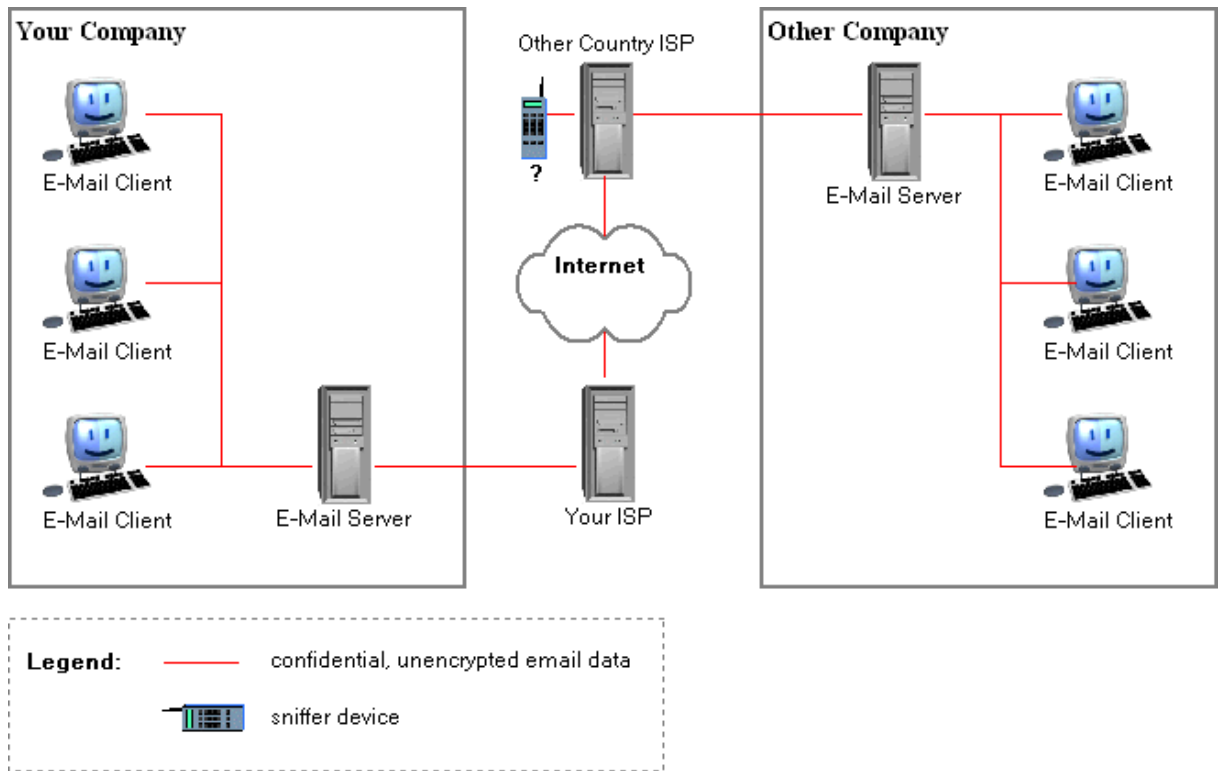
- Coexistence with other encryption standards

SecExMail Gate encrypts the mail stream and therefore does not interfere with existing methods of encryption. As such, it is possible to encrypt with PGP or GPG first, and then send the resulting cipher text through SecExMail Gate for further encryption. On the remote end, the recipients SecExMail Gate restores the PGP cipher text which can then be decrypted by the user's email client or associated PGP decryption module.

## 1.2 Overview

### Before SecExMail

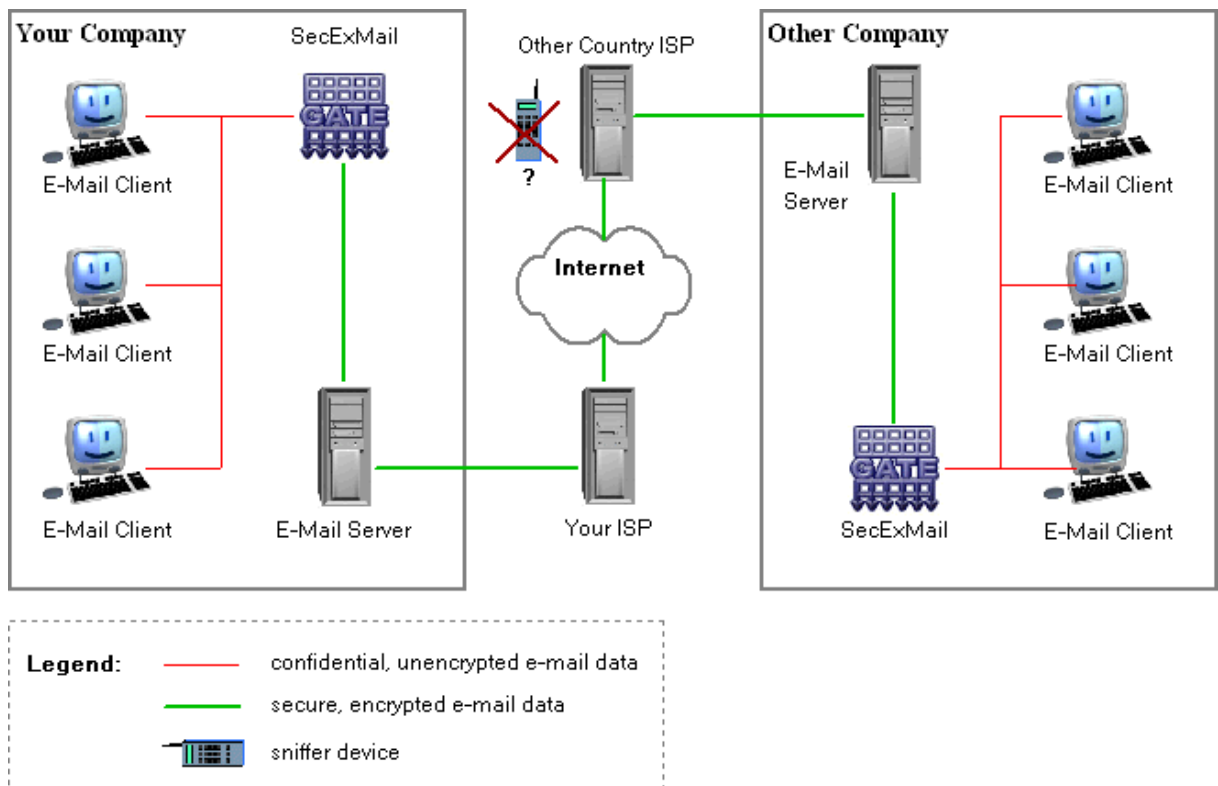
The diagram below illustrates the conventional flow of email data across corporate, local area networks and the public internet. Traditionally, email messages travel across the internet unencrypted, not unlike messages on postcards. And just like postcards, everybody involved in the delivery of an e-mail message can read it without "opening the envelope" and thus without leaving any indication that the confidentiality of the message has been compromised. As the global economy moves into cyberspace, an ever increasing volume of email traffic includes information of industrial intelligence value. This information is routinely harvested by sniffer devices either at your ISP or at key routers on the internet, scanning millions of e-mails per second. If you are communicating information of commercial value, you should take active steps now to protect your communication against industrial espionage.



### With SecExMail

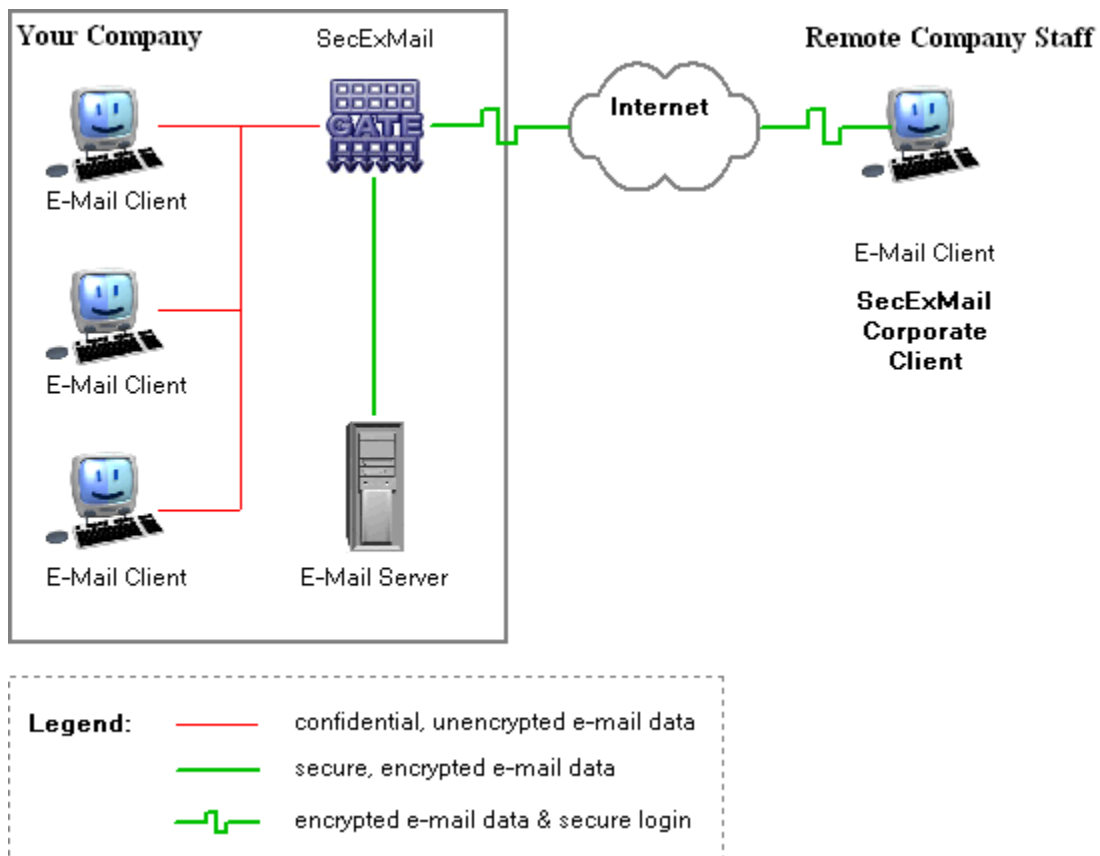
Fielding corporate e-mail encryption to protect your corporate assets is now easier than ever :

- Install the SecExMail Gate proxy service on a Windows 2000/NT/XP server on your network.
- Define at least one encryption key for your corporate domain. See [Personal and Department Keys](#)<sup>[17]</sup>.
- Configure existing email client software to communicate via SecExMail Gate
- Configure SecExMail Gate to communicate with your existing e-mail server. See [POP3](#)<sup>[17]</sup> & [SMTP](#)<sup>[8]</sup>.



### Remote Staff and Home Users

If you employ tele-commuting staff or other remote staff, you can give them direct access to your SecExMail Gate server using the corporate edition of the SecExMail client software. Remote staff can install SecExMail Corporate edition on portable notebook computers and perform encryption / decryption locally while also protecting login and account information. See [SecExMail POP3 clients - Internet](#)<sup>[12]</sup> and [SecExMail SMTP clients - Internet](#)<sup>[10]</sup>. A free edition of the SecExMail client software is available to home users if you need to communicate securely with individuals outside of your own organization.

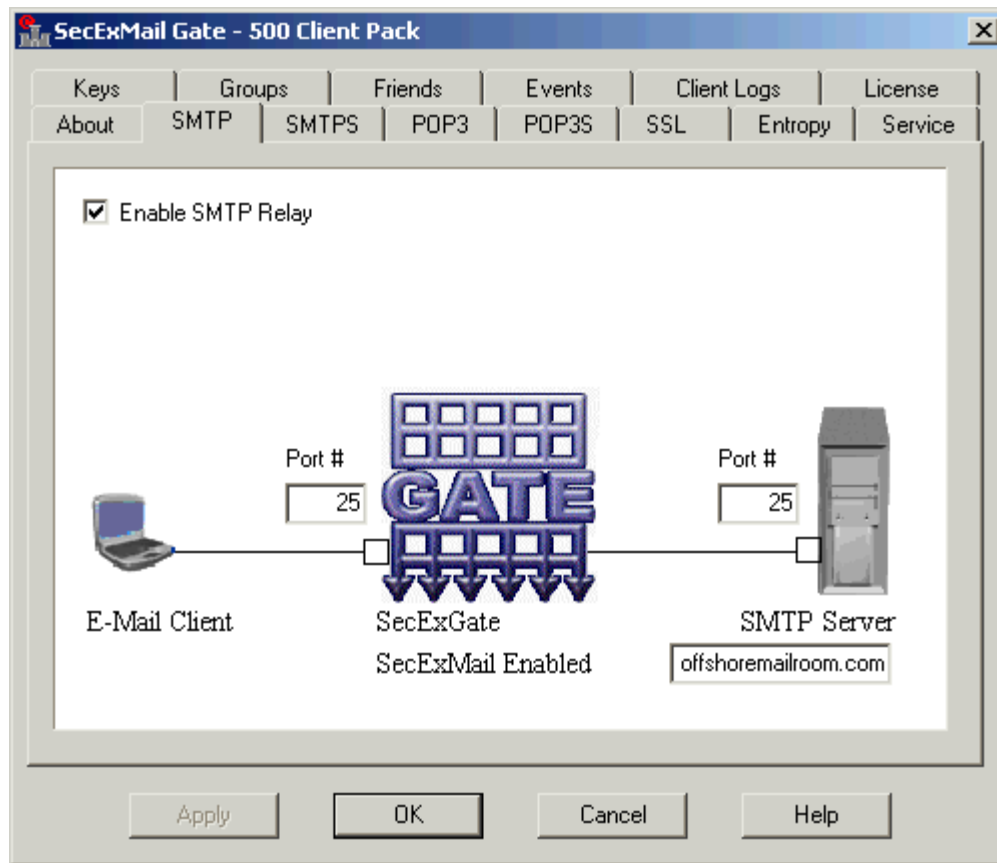


## 2 Configuration

### 2.1 Conventional SMTP clients - LAN

The SMTP tab allows you to configure SecExMail Gate for operation with mail clients on your local area network. This configuration is recommended where no secure communication between the email client workstation and your mail server is necessary. SecExMail message encryption and decryption takes place on the computer hosting SecExMail Gate. No encryption software of any kind is required on individual email client workstations. Encryption keys are managed centrally on the computer hosting SecExMail Gate by the administrator.





- **Enable SMTP Relay**

If checked, this setting enables the SMTP relay service. Message encryption is performed by SecExMail Gate. No software other than a conventional, SMTP capable email client is required on each workstation.

- **SecExGate Port**

SecExMail Gate will listen for client connections on the port you specify here. E-Mail clients should be configured to send mail via the IP address of the computer hosting SecExMail Gate using this port.

- **SMTP Server Port**

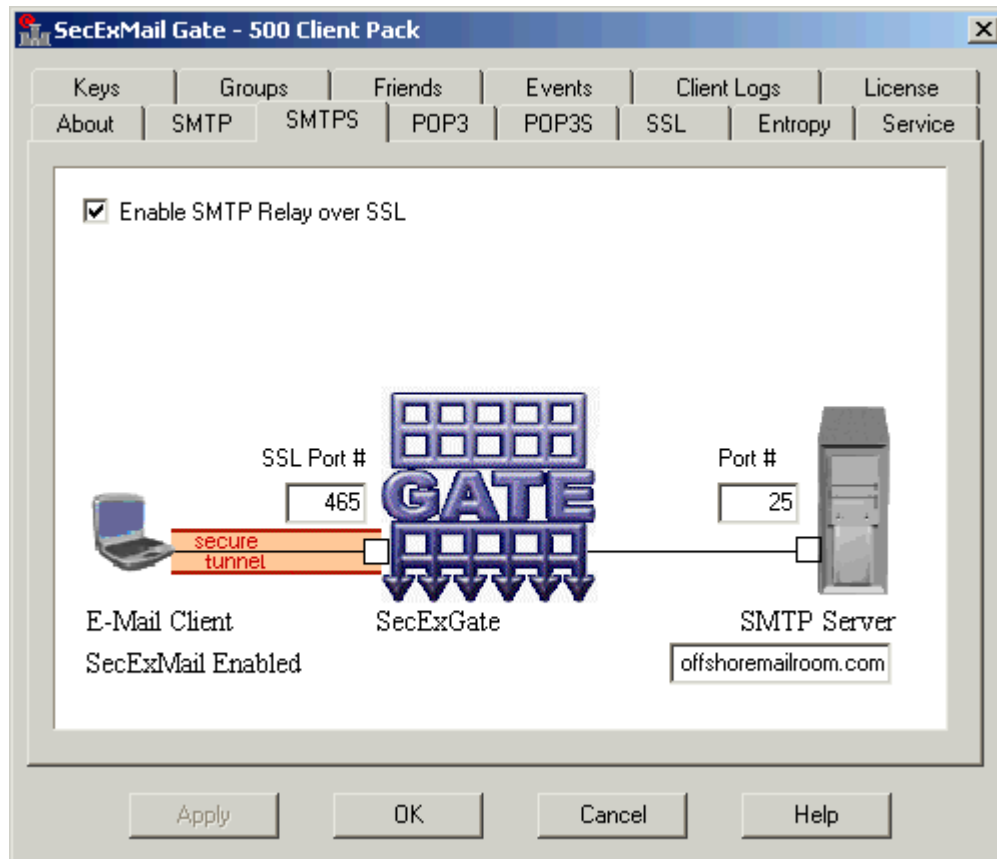
SecExMail Gate will forward e-mail data to your SMTP server on the port you specify using this setting.

- **SMTP Server**

Use this setting to specify the IP address or DNS name of your SMTP e-mail server.

## 2.2 SecExMail SMTP clients - Internet

The SMTPS tab allows you to configure SecExMail Gate for operation with SecExMail corporate email clients. This configuration option is recommended for "*road warriors*" or remote staff who will be connecting to your corporate email server from the public internet. All data and password information is protected in transit between remote staff and your corporate network by a Secure Socket Layer (SSL) tunnel. SecExMail message encryption and decryption takes place on the road warrior's computer using his/her keys. Unencrypted messages and encrypted messages alike travel securely through the SSL tunnel.



- **Enable SMTP Relay over SSL**

If checked, this setting enables the secure SMTP relay service. Message encryption is performed by SecExMail on the user's workstation. SecExMail Gate acts as secure conduit between the remote user and the corporate network.

- **SecExGate Port**

SecExMail Gate will listen for client connections on the port you specify here. E-Mail clients should be configured to send mail via the IP address of the computer hosting SecExMail Gate using this port.

- **SMTP Server Port**

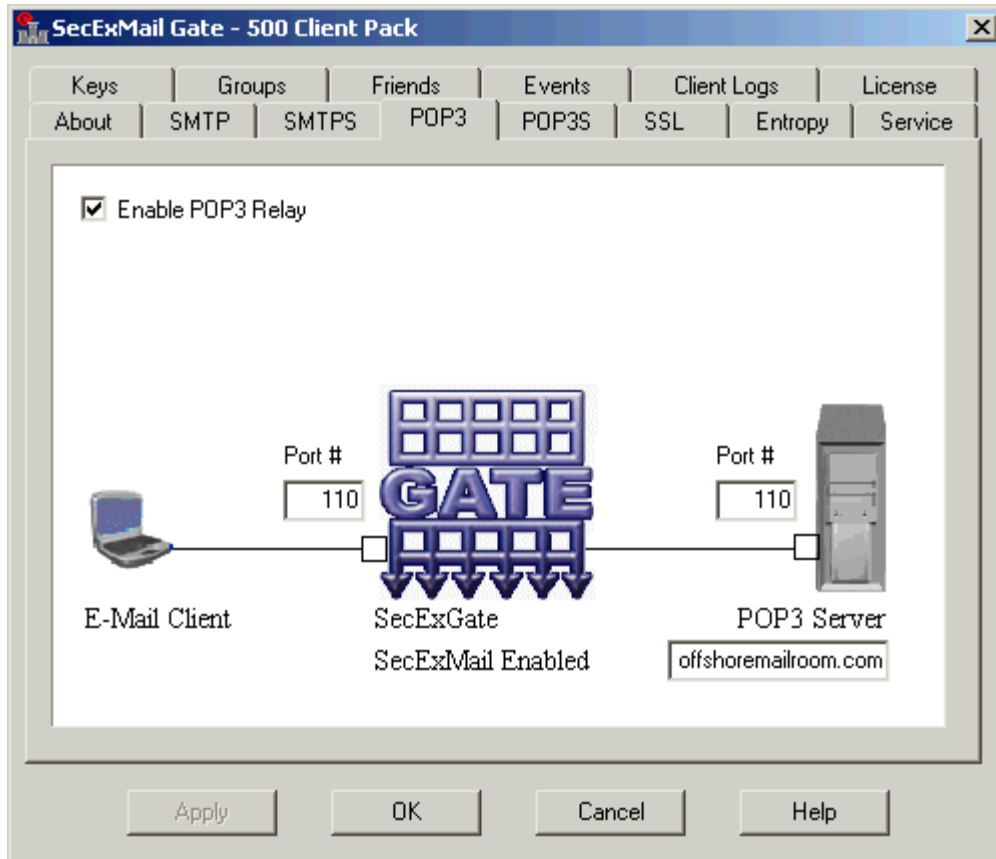
SecExMail Gate will forward e-mail data to your SMTP server on the port you specify using this setting.

- **SMTP Server**

Use this setting to specify the IP address or DNS name of your SMTP e-mail server.

## 2.3 Conventional POP3 clients - LAN

The POP3 tab allows you to configure SecExMail Gate for operation with mail clients on your local area network. This configuration is recommended where no secure communication between the email client workstation and your mail server is necessary. SecExMail message encryption and decryption takes place on the computer hosting SecExMail Gate. No encryption software of any kind is required on individual email client workstations. Encryption keys are managed centrally on the computer hosting SecExMail Gate by the administrator.



- **Enable POP3 Relay**

If checked, this setting enables the POP3 relay service. Message decryption is performed by SecExMail Gate. No software other than a conventional, POP3 capable email client is required on each workstation.

- **SecExGate Port**

SecExMail Gate will listen for client connections on the port you specify here. E-Mail clients should be configured to receive mail via the IP address of the computer hosting SecExMail Gate using this port.

- **POP3 Server Port**

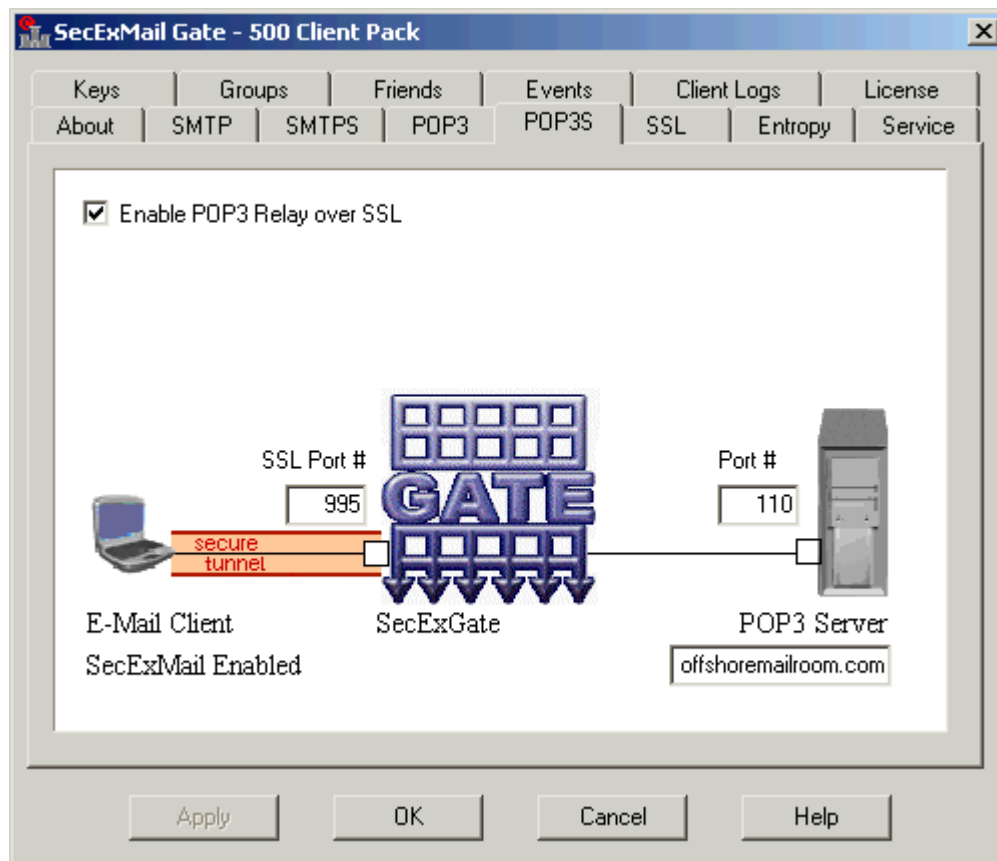
SecExMail Gate will receive e-mail data from your POP3 server on the port you specify using this setting.

- **POP3 Server**

Use this setting to specify the IP address or DNS name of your POP3 e-mail server.

## 2.4 SecExMail POP3 clients - Internet

The POP3S tab allows you to configure SecExMail Gate for operation with SecExMail corporate email clients. This configuration option is recommended for "*roadwarriors*" or remote staff who will be connecting to your corporate email server from the public internet. All data and password information is protected in transit between remote staff and your corporate network by a Secure Socket Layer (SSL) tunnel. SecExMail message encryption and decryption takes place on the road warrior's computer using his/her keys. Unencrypted messages and encrypted messages alike travel securely through the SSL tunnel.



- **Enable POP3 Relay over SSL**

If checked, this setting enables the secure POP3 relay service. Message decryption is performed by SecExMail on the user's workstation. SecExMail Gate acts as secure conduit between the remote user and the corporate network.

- **SecExGate Port**

SecExMail Gate will listen for client connections on the port you specify here. E-Mail clients should be configured to receive mail via the IP address of the computer hosting SecExMail Gate using this port.

- **POP3 Server Port**

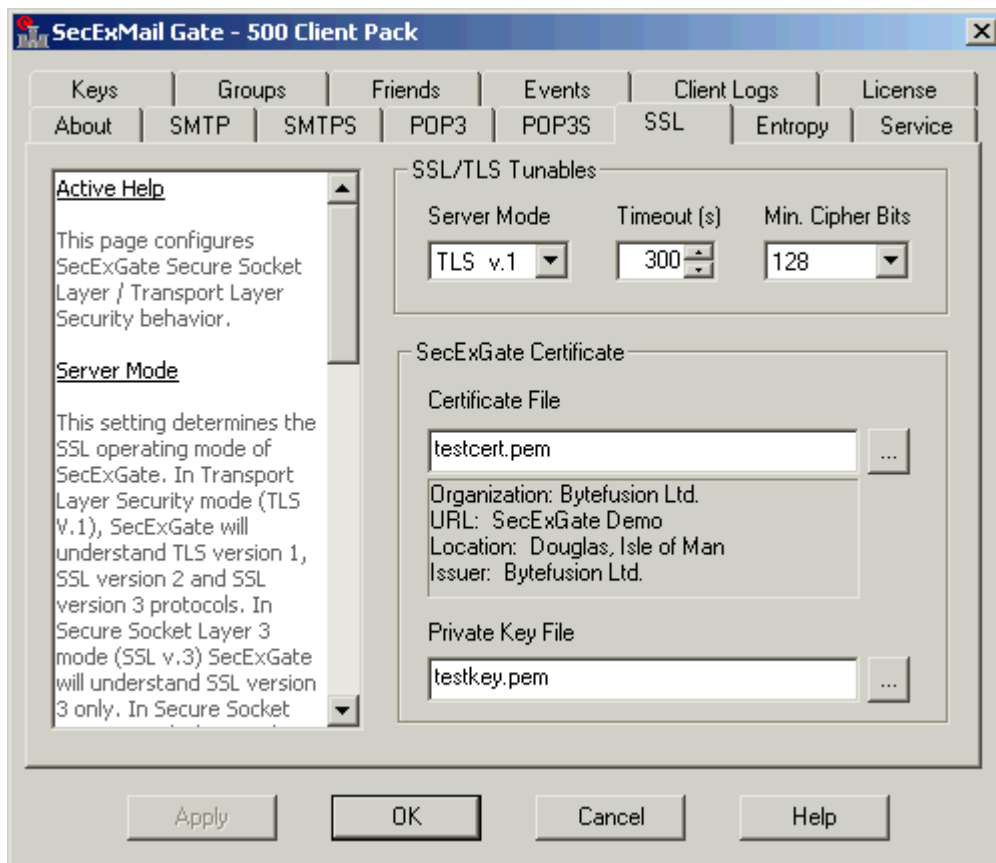
SecExMail Gate will receive e-mail data from your POP3 server on the port you specify using this setting.

- **POP3 Server**

Use this setting to specify the IP address or DNS name of your POP3 e-mail server.

## 2.5 Secure Socket Layer & Certificates

The Secure Socket Layer (SSL) page allows you to fine tune the operation of secure client connections and configure the SSL certificate SecExMail Gate will present to connecting clients. This certificate serves to verify the identity of the computer hosting SecExMail Gate to clients connecting from outside your corporate network and thus prevents Trojan horse attacks which could otherwise compromise sensitive password information. See also [IP/DNS spoofing](#)<sup>46</sup>.



- **Server Mode**

This setting determines the SSL operating mode of SecExGate. In Transport Layer Security mode

(TLS V.1), SecExGate will understand TLS version 1, SSL version 2 and SSL version 3 protocols. In Secure Socket Layer 3 mode (SSL v.3) SecExGate will understand SSL version 3 only. In Secure Socket Layer 2 mode (SSL v.2) SecExGate will understand SSL version 2 only. For compatibility reasons, the recommended setting is "TLS V.1"

- **Timeout**

This setting determines the SSL session timeout. The recommended default is 300 seconds.

- **Min. Cipher Bits**

This setting determines the minimum encryption strength for SSL/TLS ciphers used by SecExGate. Only ciphers satisfying this requirement will be included in the SSL protocol negotiation with clients.

- **Certificate File**

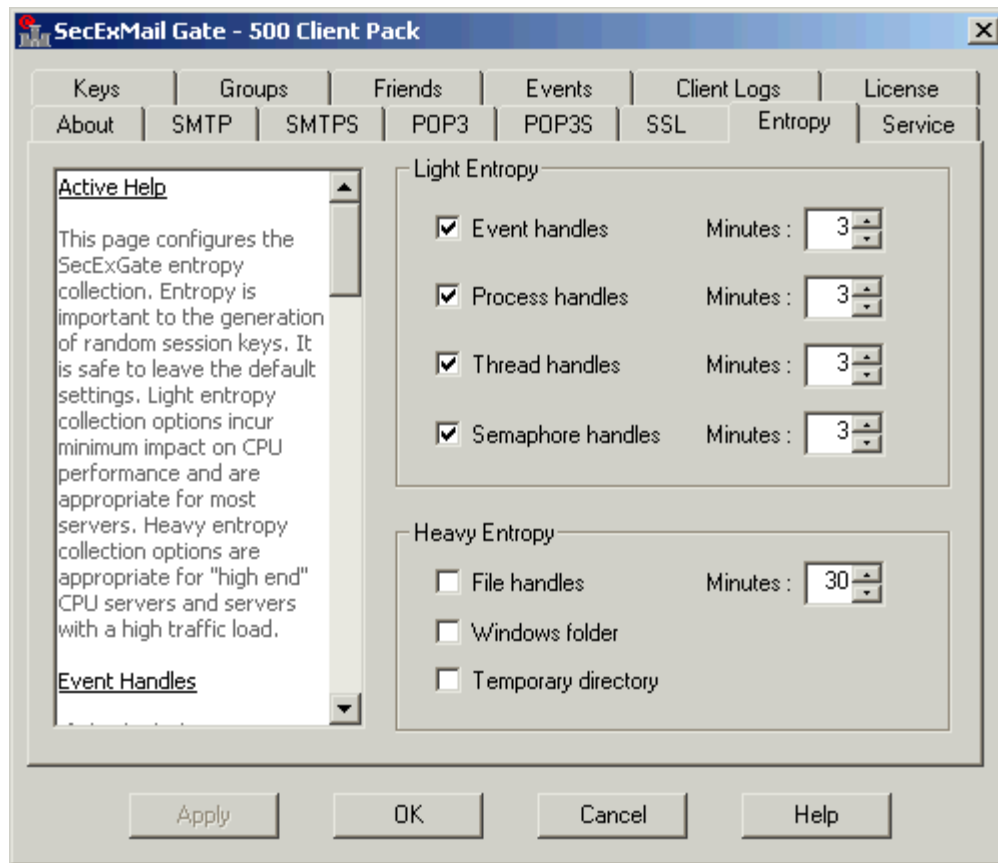
This setting should specify the full path to your SSL certificate in PEM encoded format. Certificates are used by clients to verify the identity of your SecExGate mail relay server. See also [Generating Certificate Requests](#)<sup>[28]</sup>.

- **Private Key File**

This setting should specify the full path to the RSA key for your SSL certificate. The key file must \*NOT\* be encrypted.

## 2.6 Entropy Collection Settings

This page configures the SecExMail Gate entropy collection. Entropy is important to the generation of random session keys. It is safe to leave the default settings. Light entropy collection options incur minimum impact on CPU performance and are appropriate for most servers. Heavy entropy collection options are appropriate for "high end" CPU servers and servers with a high traffic load. Ordinarily, the best source of entropy on most computer systems is the user him/herself. On server systems where no user interaction exists extra care must be taken to guarantee good entropy and, by extension, guarantee selection of encryption keys which are unpredictable to any attacker. See [Entropy Collection](#)<sup>[42]</sup>.



- **Event Handles**

If checked, this setting prompts SecExMail Gate to collect information about all event handles on this computer at the specified interval. Variations in the collected data are distilled into the entropy pool via a secure hash function.

- **Process Handles**

If checked, this setting prompts SecExMail Gate to collect information about all process handles on this computer at the specified interval. Variations in the collected data are distilled into the entropy pool via a secure hash function.

- **Thread Handles**

If checked, this setting prompts SecExMail Gate to collect information about all thread handles on this computer at the specified interval. Variations in the collected data are distilled into the entropy pool via a secure hash function.

- **Semaphore Handles**

If checked, this setting prompts SecExMail Gate to collect information about all semaphore handles on this computer at the specified interval. Variations in the collected data are distilled into the entropy pool via a secure hash function.

- **File Handles**

If checked, this setting prompts SecExMail Gate to collect information about all file handles on this computer at the specified interval. Variations in the collected data are distilled into the entropy pool via a secure hash function.

- **Windows folder**

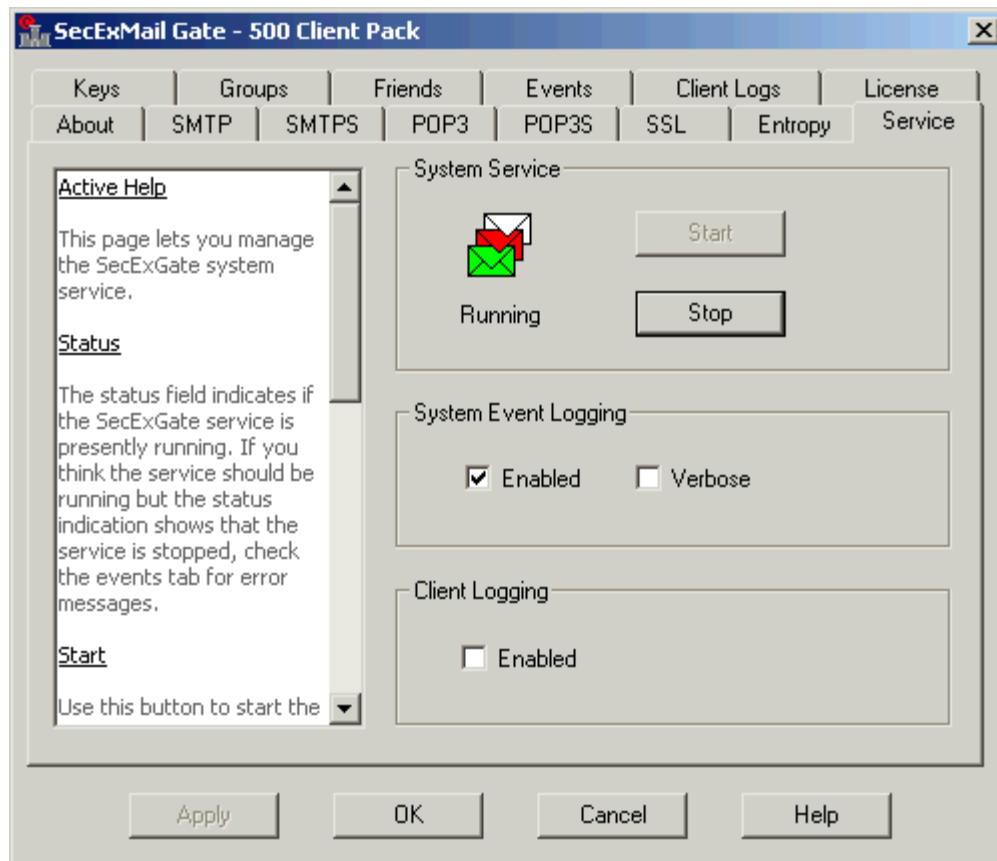
If checked, this setting prompts SecExMail Gate to register with the operating system for notification of file changes in the Windows folder. When changes are signalled, SecExMail Gate compiles file information on the Windows folder which is distilled into the entropy pool via a secure hash function.

- **Temporary directory**

If checked, this setting prompts SecExMail Gate to register with the operating system for notification of file changes in the temporary directory. When changes are signalled, SecExMail Gate compiles file information on the temporary directory which is distilled into the entropy pool via a secure hash function.

## 2.7 System Service Settings

This page lets you fine tune the operation of SecExMail Gate as a system service.



- **Status**

The status field indicates if the SecExMail Gate service is presently running. If you think the service



should be running but the status indication shows that the service is stopped, check the events tab for error messages.

- **Start**

Use this button to start the SecExMail Gate system service.

- **Stop**

Use this button to stop the SecExMail Gate system service.

- **Enable system event logging**

This setting specifies if SecExMail Gate logs information to the operating system application event log.

- **Verbose system event logging**

This setting specifies if SecExMail Gate logs verbose information to the operating system application event log.

- **Enable client logging**

This setting specifies if SecExMail Gate logs information about client sessions to file.

## 2.8 Personal and Department Keys

The keys page allows you to manage your SecExMail keys. If a key is displayed on this page, both a [public and a private key component](#)<sup>[40]</sup> is on file for corresponding email address. Messages to and from the listed email address may be encrypted as well as decrypted. Group keys defined on the [groups page](#)<sup>[19]</sup> will also be displayed on this page.

The following types of keys can be defined :

- **Wildcard keys**

A wildcard key is a generic key which represents a domain without being associated with a specific email address in the domain. A wildcard key is used to encrypt messages when sending e-mail to a member of the domain and no matching personal key is found for the member of the domain. This means two companies can secure their communication by exchanging only their domain keys.

Example: Using the key chain shown in the screenshot below, an e-mail message is sent to "joe@offshoremailroom.com". The key chain below only defines the following personal and department keys for the *offshoremailroom.com* domain : "accounting", "ceo", "marketing", "sales" and "support". Because no matching personal key or department key is found, the message is encrypted using the wildcard key for the domain.

- **Personal keys**

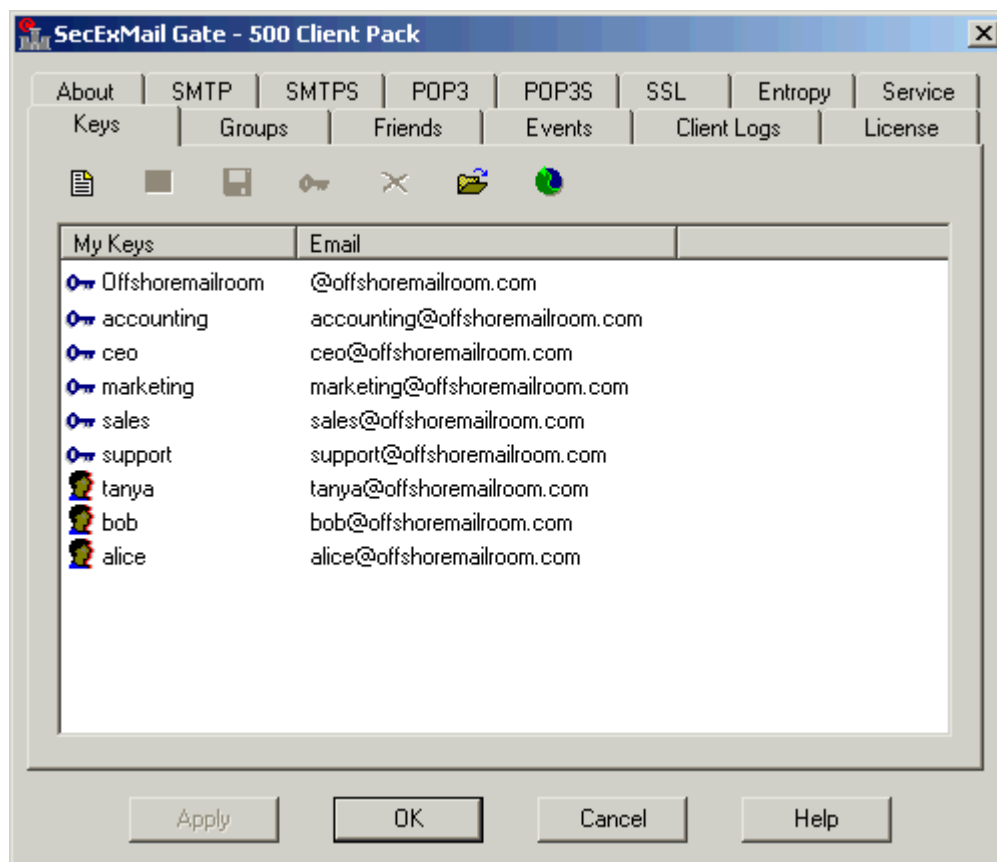
Personal keys are keys associated with the email address of an individual within a domain. The key "ceo@offshoremailroom.com" is in the screenshot below is an example of a personal key. Because keys are managed centrally by SecExGate, personal keys need not be distributed to users's workstations. In cases where users prefer to manage their own keys, they may elect to run the

corporate edition of SecExMail client directly on their respective workstations.

- **Department keys**

A department key is a key representing a department or group within an organization. Typically individuals within the organization will be assigned to a department using the [groups page](#)<sup>[19]</sup> but do not require their own keys.

Example: The user *Tanya* shown below is associated with the *accounting* group. See [groups page](#)<sup>[19]</sup> on how to create this association. In practical terms this means when the staff member Tanya sends encrypted mail, an SMTP header in the outgoing message instructs remote installations of SecExGate to use the *accounting* key for encryption when replying to Tanya. This means no physical key for the staff member Tanya needs to be created.



- **New key button** 

Launch the SecExMail key generator to add a new SecExMail key to your key chain.

- **Key properties button** 

Display detailed information about the selected key

- **Export Key button** 

Save the selected key to disk

- **Change key passphrase button** 

Change the passphrase which protects the selected key as stored in the registry

- **Delete key button** 

Delete the selected key from the registry

- **Import key button** 

Open a new key as stored on disk and save it in the registry

- **Re-read keys button** 

You may have to refresh the display by clicking this button after manually editing configuration files such as **groupkeys.csv**.

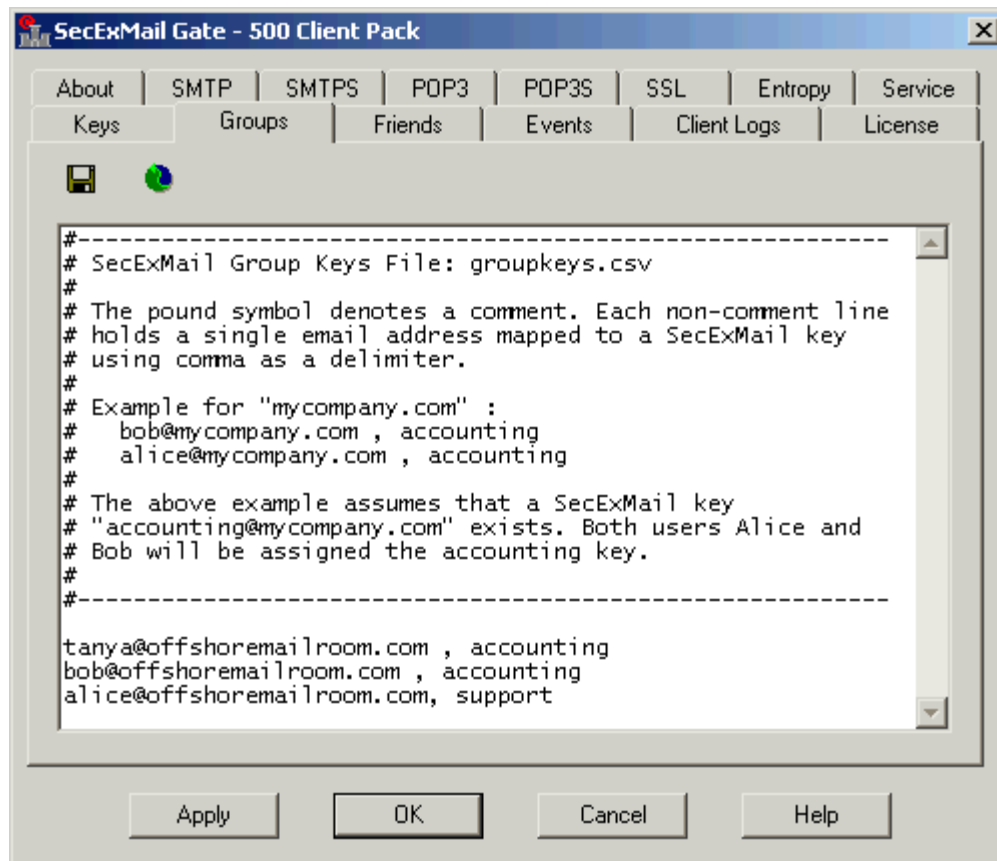
## 2.9 Group Key Management

The group key management page is used to define associations between individual staff members and their respective departments. To provide interoperability with third party applications such as spreadsheets and database programs, group key associations are maintained in a comma separated values file, "**groupkeys.csv**", in the application folder of SecExMail Gate. You may edit details of group key associations manually on this page or export your staff database directly from your spreadsheet or database application. The format of the file groupkeys.csv is defined as follows :

The pound symbol denotes a comment. Each non-comment line holds a single email address mapped to a SecExMail key using comma as a delimiter. Empty lines are ignored.. Example for "mycompany.com" :

```
bob@mycompany.com , accounting
alice@mycompany.com , accounting
```

The above example assumes that a physical SecExMail key "accounting@mycompany.com" is defined for the accounting department on the [keys page](#)<sup>[17]</sup>. Both users Alice and Bob will be assigned the accounting key.



- **Save changes button** 

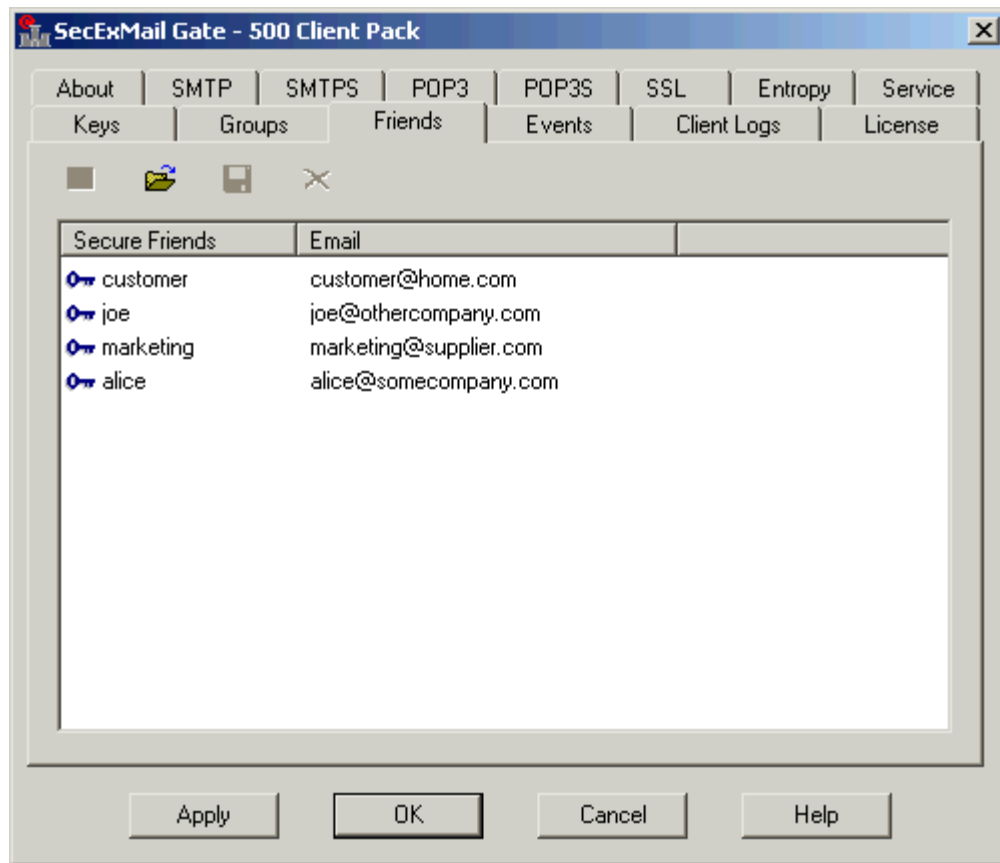
Save your changes back to the file "**groupkeys.csv**"

- **Re-read groups file** 

You may have to refresh the display by clicking this button after modifying the **groupkeys.csv** file outside of SecExMail Gate such as may be the case when exporting data from a database program or spreadsheet application.

## 2.10 Friend Keys Management

The Friends page allows you to manage SecExMail keys of friends and business partners. If a key is displayed on this page, only a [public key component](#)<sup>[40]</sup> is on file for corresponding email address. Messages may be encrypted to the listed email address but only the holder of the private key component may decrypt the message.



- **Key properties button** 

Display detailed information about the selected key

- **Import key button** 

Open a new key as stored on disk and save it in the registry

- **Export Key button** 

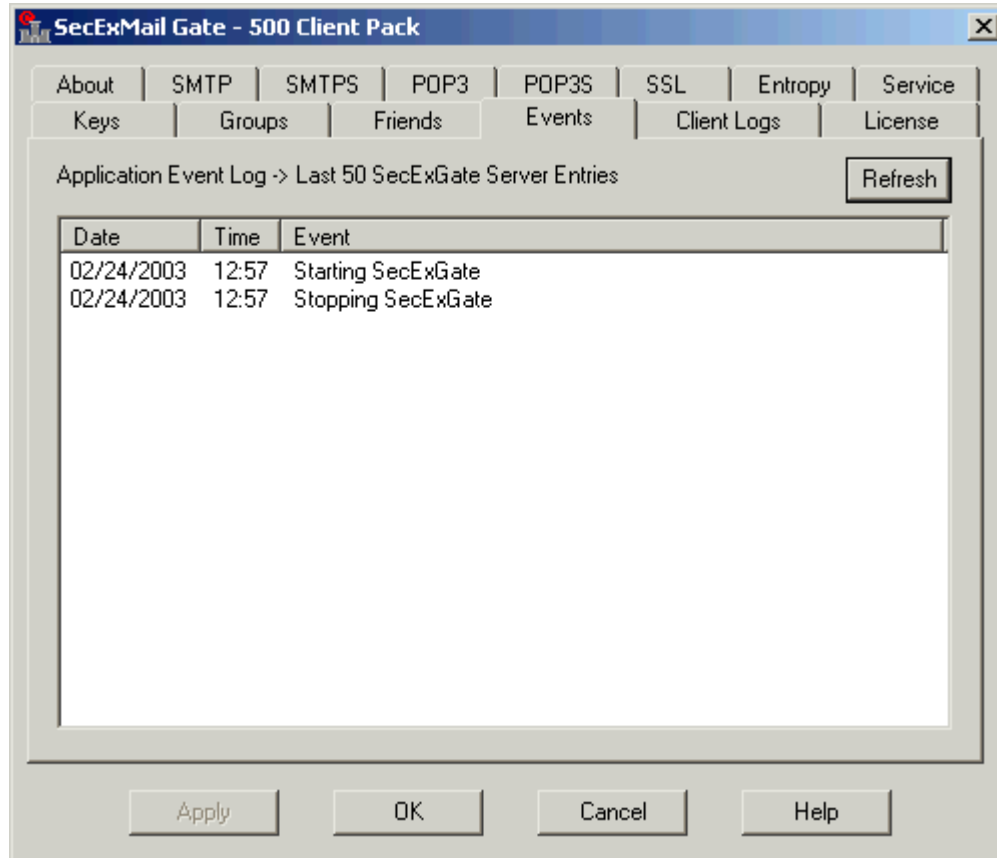
Save the selected key to disk

- **Delete key button** 

Delete the selected key from the registry

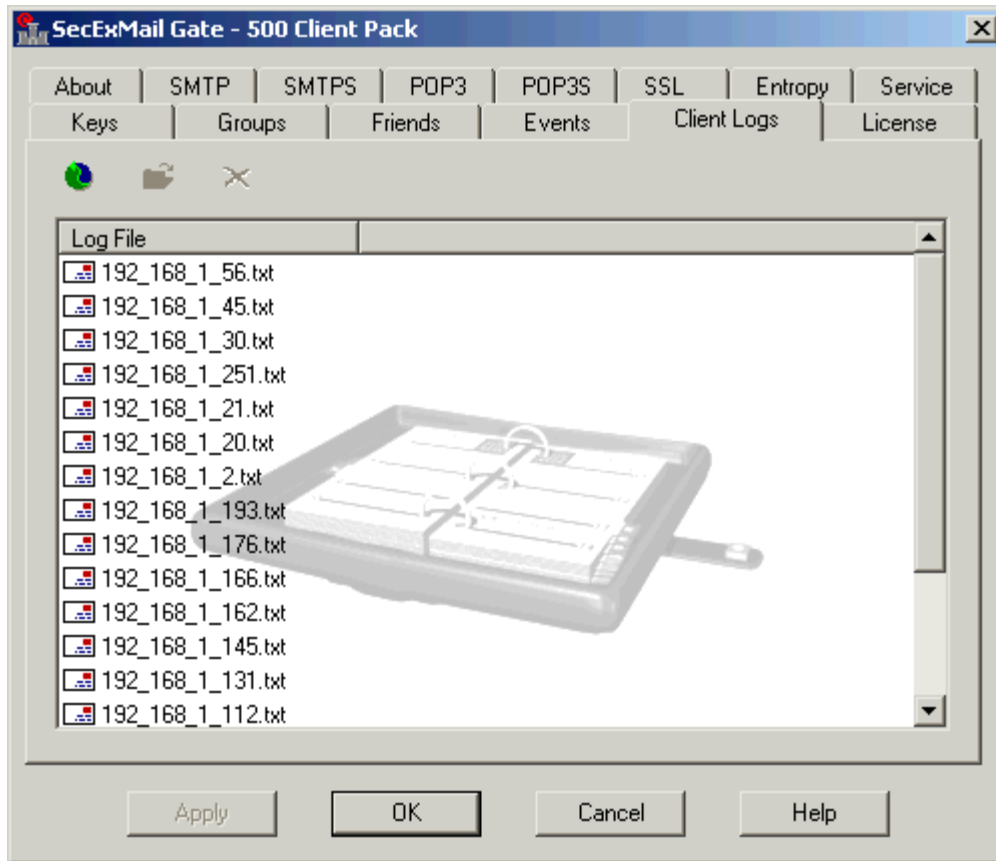
## 2.11 System Event Log

The system event log page displays the last 50 messages SecExMail Gate has reported to the operating system event log. Hit the **Refresh** button to update the display.

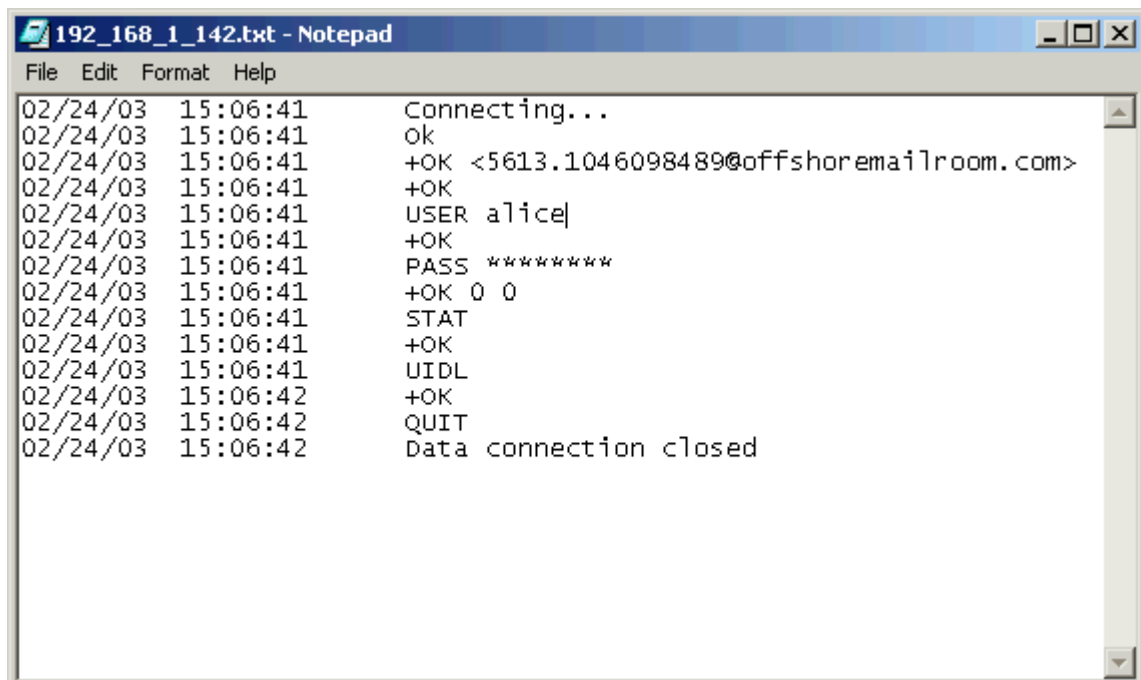


## 2.12 Client Event Logs

Client event logs can be used to debug problems between your e-mail clients, SecExMail Gate and your mail server. In order to record client logs, you will first need turn client logging on via the [Events tab](#). It is recommended that client logging only be performed for debugging purposes as the amount of disk space consumed by client logs can grow rapidly. Client event logs are created only for local clients, not for road warriors operating their own version of SecExMail client, corporate edition. SecExMail clients record their own debugging information.



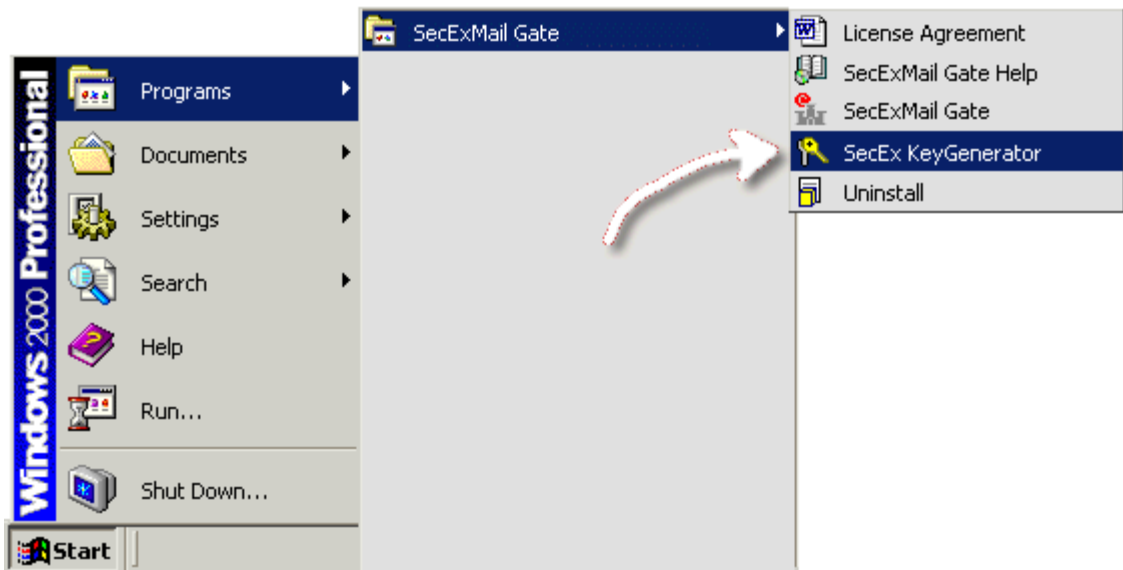
Client logs do not record the content of users' emails nor do they record users' password information. Client logs merely record the history of client commands and server replies and associated SecExMail messages. A typical client log is shown below.



## 3 Keys

### 3.1 Create your personal SecExMail keys

The SecEx Key Generator creates [encryption keys](#)<sup>[40]</sup> for you which will enable your friends to send you encrypted mail and enable you to decrypt mail sent to you by your friends. To invoke the SecEx Key Generator, click "Start", "Programs", "SecExMail Gate" and "SecEx KeyGenerator" as shown below.



This will start the SecEx Key Generator which will guide you through the process of creating your own SecExMail keys.






Click **Next** to proceed to the [Personal Details](#) screen.

## 3.2 Personal Details Screen

The **Personal Details** screen collects information about you and your email address. This information will later appear on the [Keys tab](#) in SecExMail. SecEx Key Generator will not disclose your information to anyone and you control whom you share your key information with.



SecEx Key Generator - Personal Details

Please provide the name and email address that will be associated with this keypair.

Your name  
dodo bird

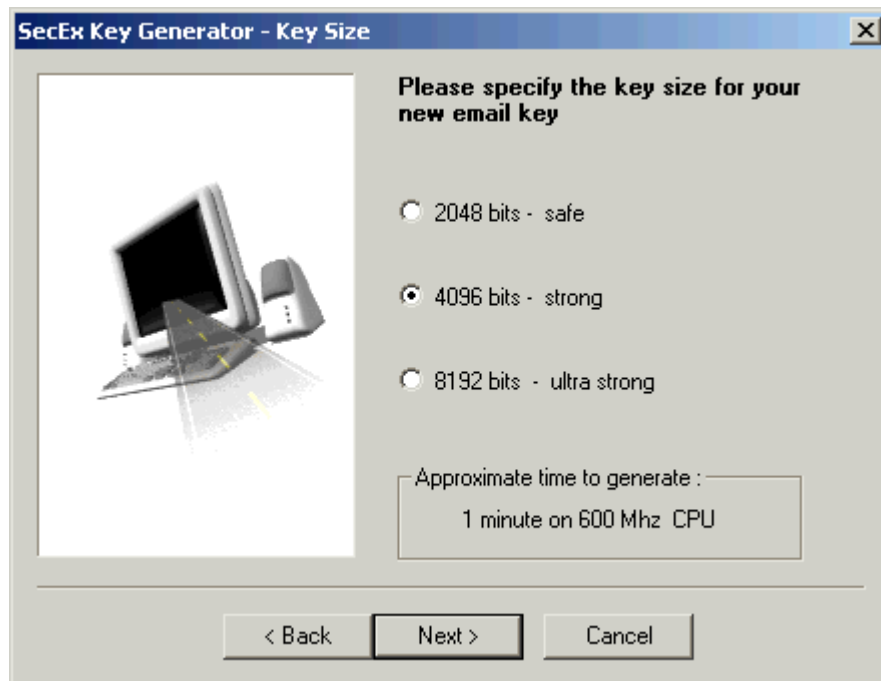
Email address  
dodo@offshoremailroom.com

< Back    Next >    Cancel

Click **Next** to go to the [Key Size](#) screen.

## 3.3 Key Size Screen

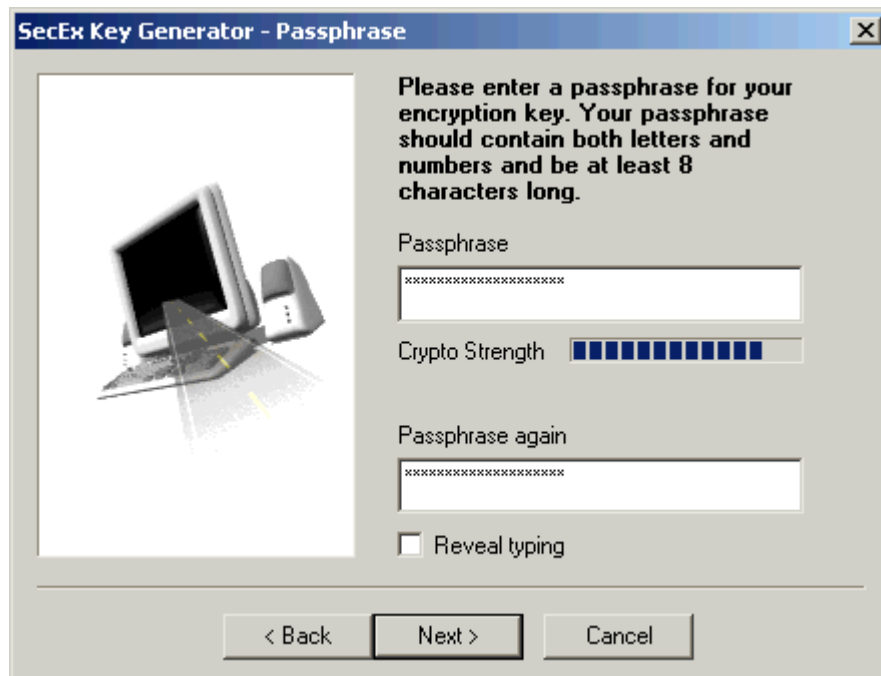
Deciding on the key size of your new email key is a matter of personal judgement. It is commonly held that RSA keys of 1024 bits will withstand conventional cryptanalytic attacks while key sizes of 512 bits or less are to be regarded as insecure. In general, the cryptographic community is divided over the recommended key size for [asymmetric keys](#) and what is referred to as the "huge key debate". If 1024 bit keys are secure, why use larger keys? The counter argument is that the only disadvantage to using larger keys is the longer time required to process them. CPU cycles are cheap and getting cheaper every year. This aids the cryptographer and cryptanalyst alike. So why not use the largest key size contemporary computers can handle? The decision is yours ...



Click **Next** to proceed to the [Passphrase](#)<sup>[26]</sup> screen.

### 3.4 Passphrase Screen

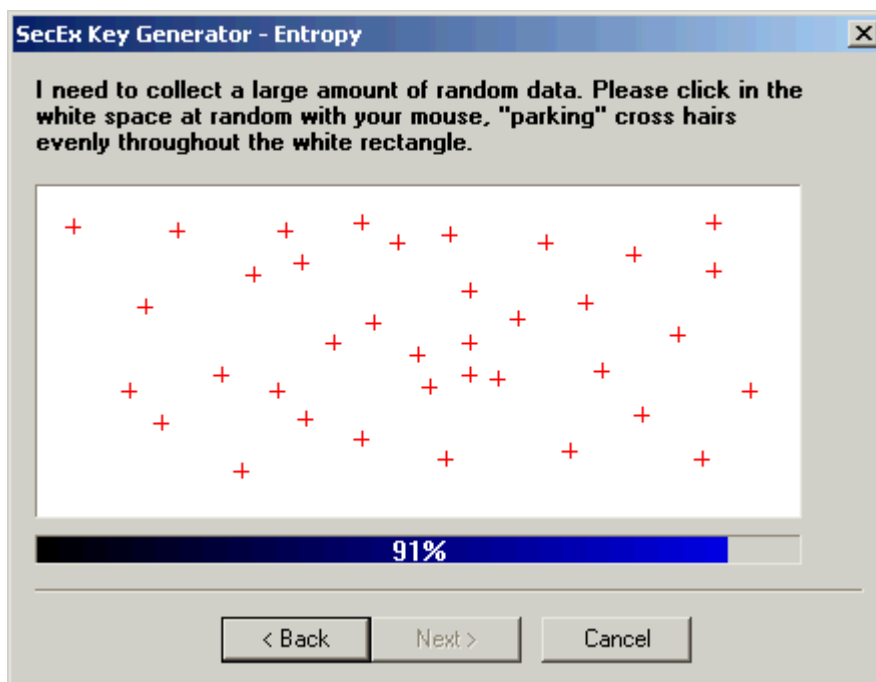
Your new key will be stored in your computer's registry. To protect confidential key information from unauthorized access, it will be [encrypted and protected](#)<sup>[40]</sup> with a passphrase that only you know. Please choose a long phrase containing both letters and numbers and avoid using the names of girlfriend, wife, boyfriend or husband. Do not use your date of birth.



Click **Next** to proceed to the [Entropy](#) screen.

### 3.5 Entropy Screen

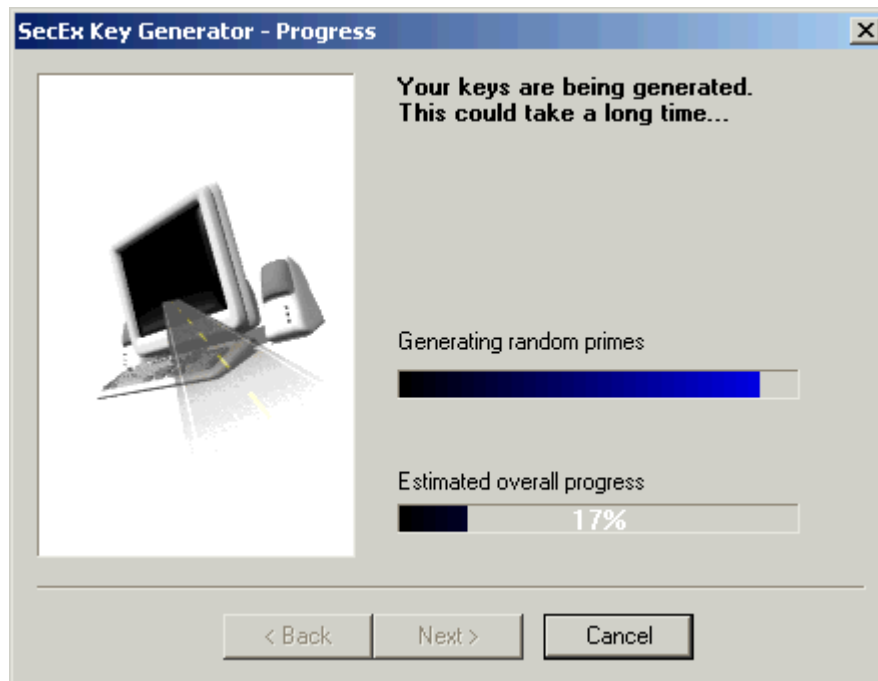
Your new key will be generated from prime numbers produced by a random number generator. In order for your key to be unpredictable we need to collect a large amount of random data from the only source in the system which is unique : **you**. This data will be used to seed the random number generator. To ensure maximum security, the SecEx Key Generator does not avail itself of previously calculated prime numbers or so called "canned primes." All prime numbers are generated "on the spot".



Click **Next** to proceed to the [Progress](#) screen.

### 3.6 Progress Screen

The progress screen shows the estimated time to completion. The estimated time to finish may be readjusted during key generation and you will be advised when key generation is complete. At that time, click **Finish** to save your keys and restart SecExMail if necessary.



## 4 Certificates

### 4.1 Generating Certificate Requests

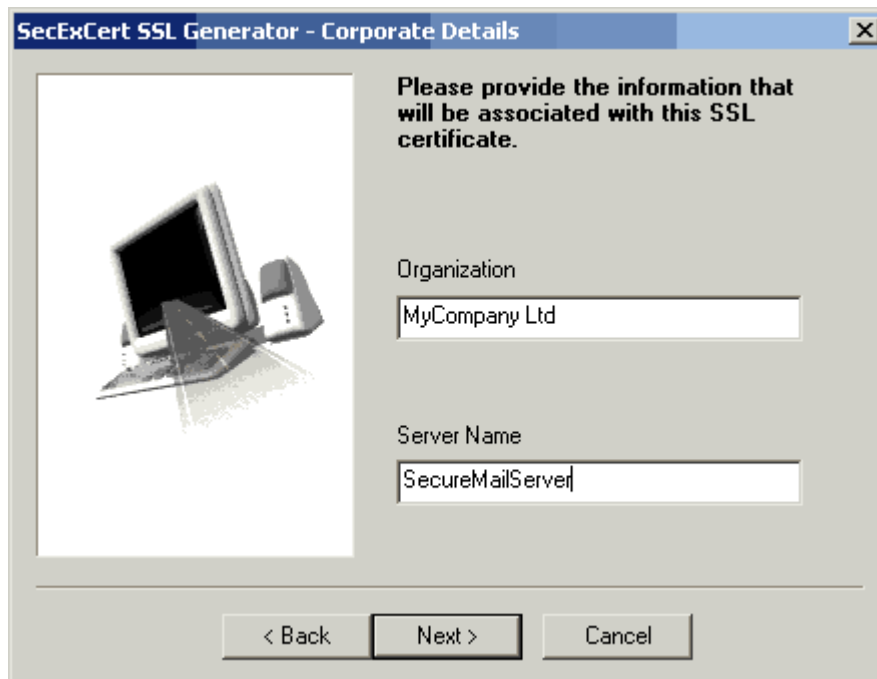
Like other SSL based servers, SecExMail Gate uses certificates to authenticate itself to clients, thus preventing breaches of security such as Trojan Horse attacks and [DNS spoofing](#)<sup>[46]</sup>. SSL certificates are based on a chain of trust which requires them to be signed by a certificate authority. Initially, SecExMail Gate ships with a self signed demo certificate for evaluation purposes. All demo certificates are identical and represent an imaginary server. When you have completed evaluation of the software you will need to generate your own certificate representing your organization. This is done by first generating a certificate request using the details of your organization. When you have generated your certificate request, you may forward the certificate request file to [support@bytefusion.com](mailto:support@bytefusion.com) for free signing. Signing or certification of certificate requests is free of charge to all end users who have acquired a licensed copy of SecExMail Gate. See also [Secure Socket Layer & Certificates](#)<sup>[13]</sup>.

Select "**Start**", "**Programs**", "**SecExMail Gate**" & "**Generate SSL Certificate Request**" to start the SecExCert certificate generator. The screenshots below illustrate the process of generating certificate requests.

#### Welcome Page




### Corporate Details Page



### Geographic Location Page

**SecExCert SSL Generator - Geographic Location**

Please enter the details of your geographical location.



City  
Douglas


Country  
Isle of Man

< Back   Next >   Cancel

### RSA Key Size Page

**SecExCert SSL Generator - Key Size**

Please specify the key size for your new SSL certificate



1024 bits

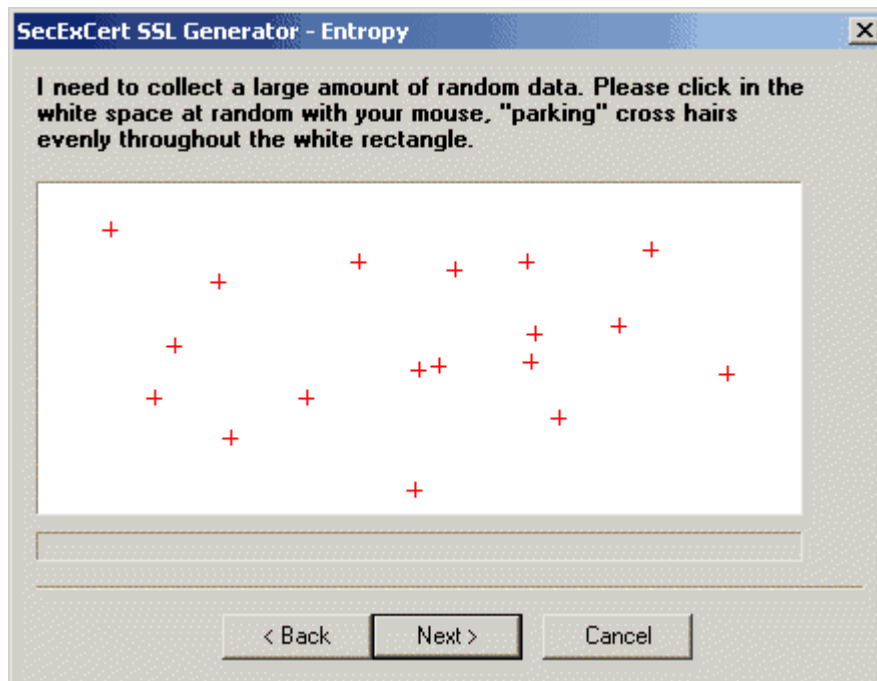
2048 bits

4096 bits

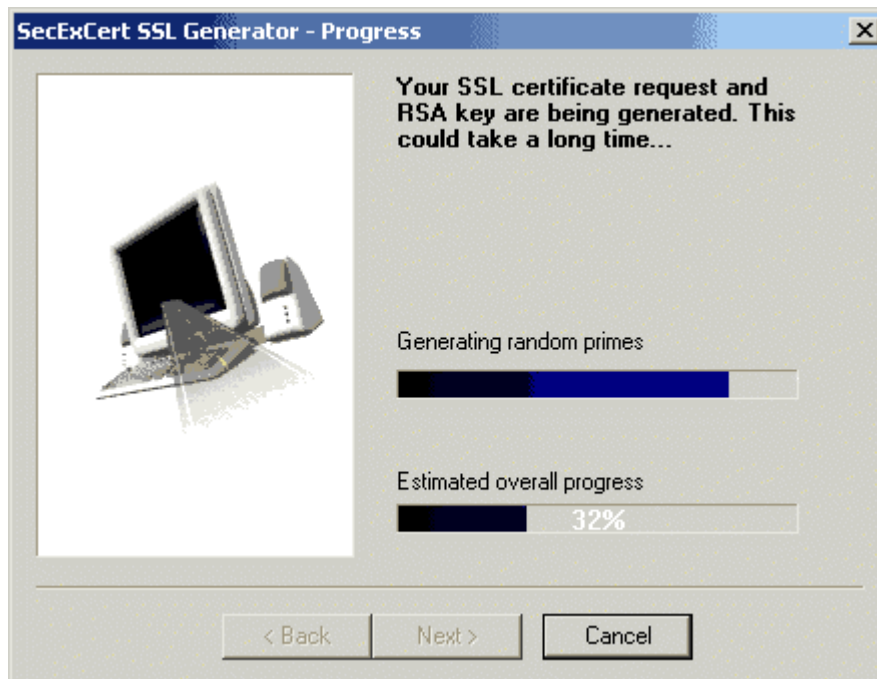
Approximate time to generate :  
1 minute on 600 Mhz CPU

< Back   Next >   Cancel

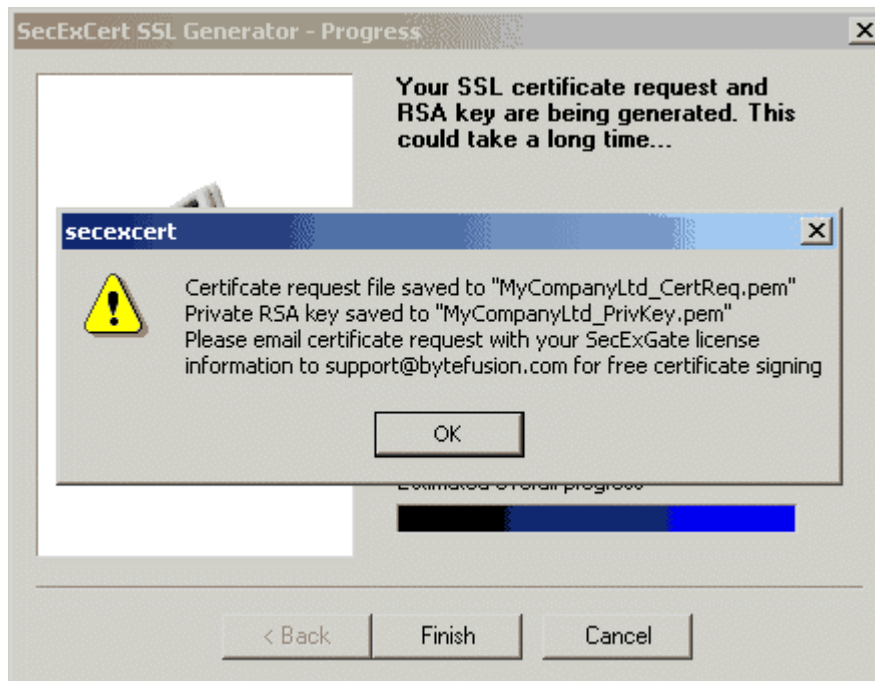
### Entropy Collection Page



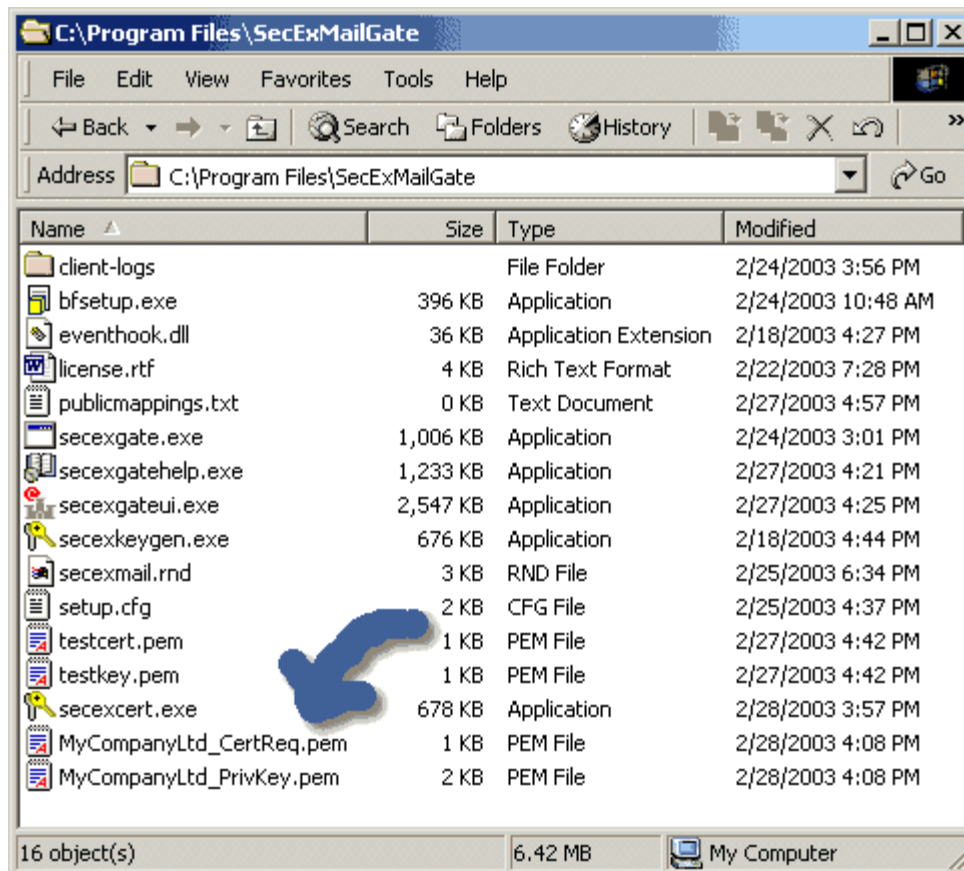
### Progress Page



### Certificate Generation Complete

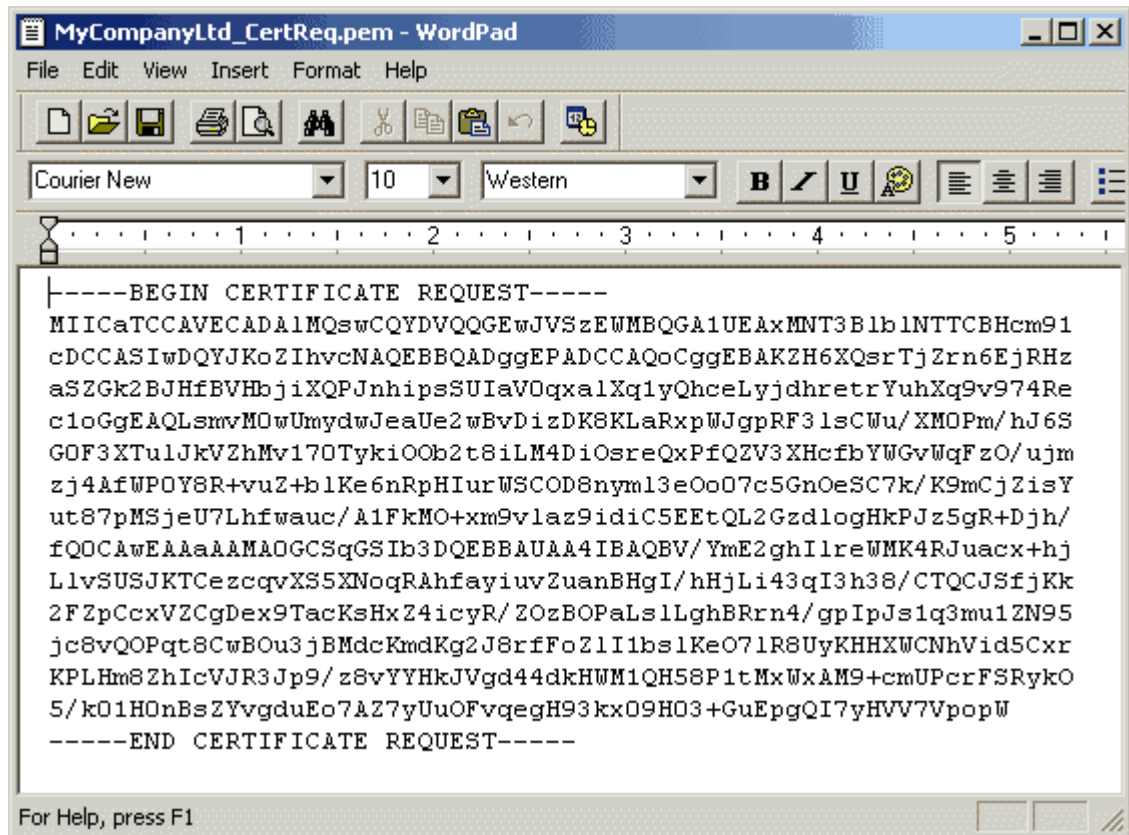


### Default Location of Generate Files



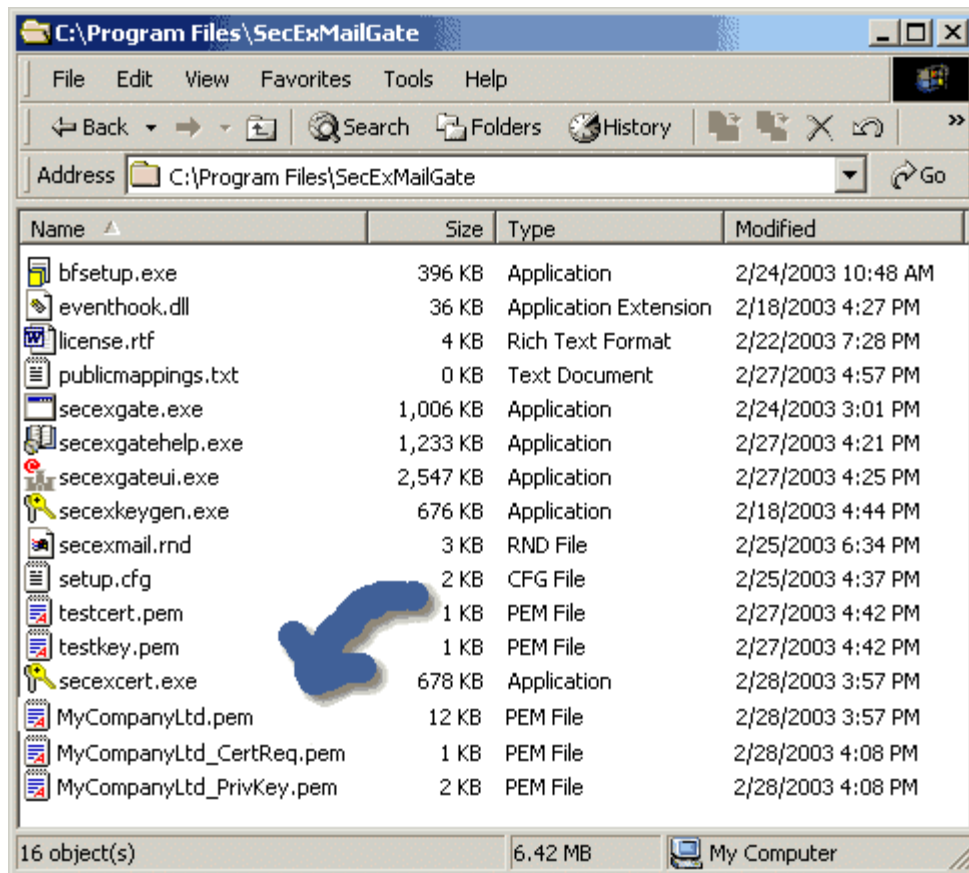


**Sample Certificate Request File in PEM Encoded Format**



**4.2 Installing Certificates**

After you have received your signed SSL certificate, you need to copy the certificate file to the application folder of SecExMail Gate as shown below. You may now delete the "???" *CertReq.pem* certificate request file. Then update the "**certificate file**" and "**private key file**" parameters on the [Secure Socket Layer & Certificates](#)<sup>[13]</sup> page and restart the SecExMail Gate service.

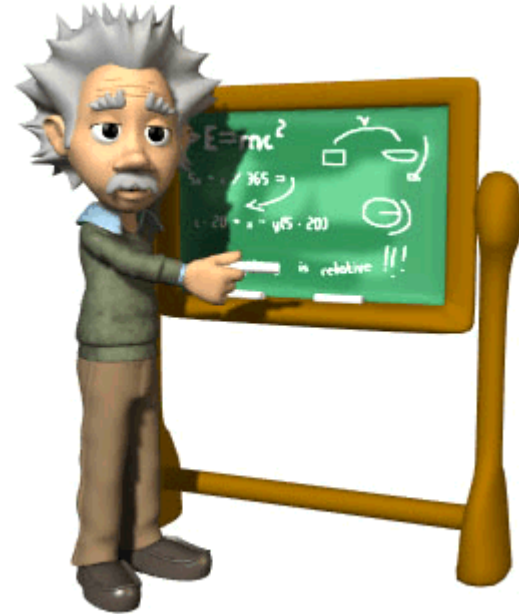


## 5 Technical

### 5.1 RSA Public Key Encryption

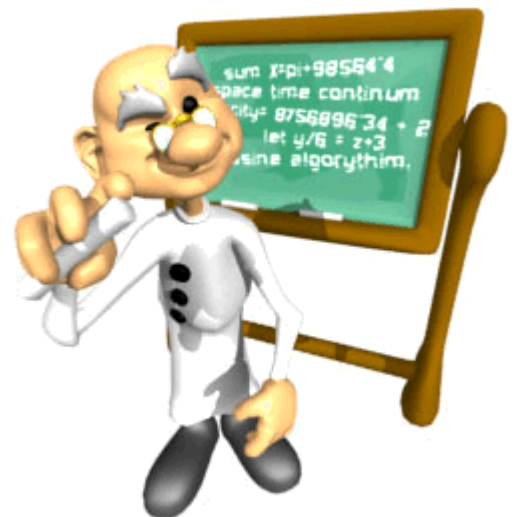
" $c = me \text{ mod } n$ " is the algorithm that turns the world of e-commerce. Introduced in 1978 by Rivest, Shamir and Adleman after whom the cipher is named, RSA is the worlds foremost public key encryption system. Contrary to the design of classic encryption algorithms where the same key is used to lock and unlock the information, public key encryption relies on "two key" algorithms. The sender encrypts the message with the recipients public key who, upon receipt of the message, is able to decipher the same with the private key counterpart. This development was revolutionary in the field of cryptography because parties wishing to establish secure communications no longer had to meet in "secret" to exchange confidential keying information.

The [SecExMail public key](#)<sup>[40]</sup> infrastructure uses industry standard RSA encryption as developed by the OpenSSL project. See [Acknowledgements](#)<sup>[50]</sup>.

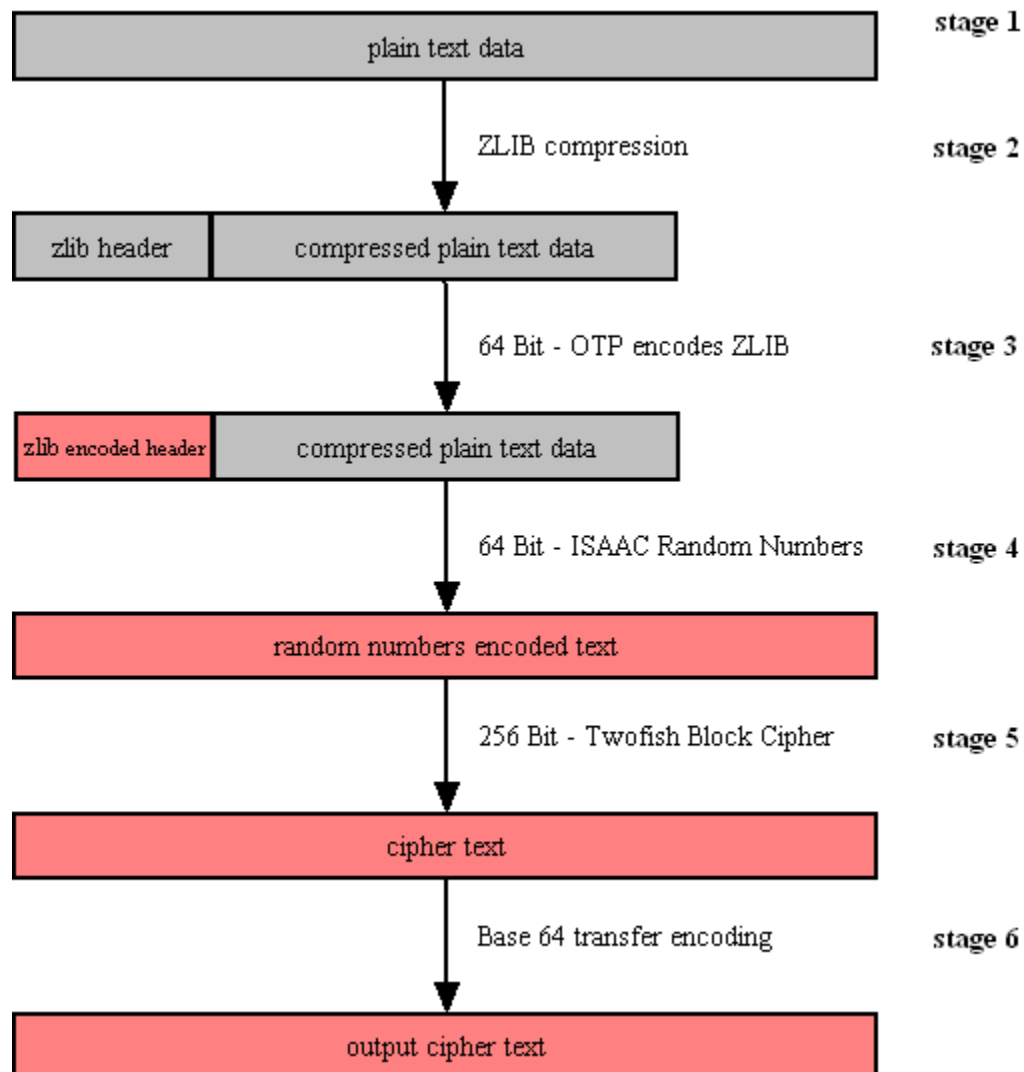


### 5.2 SecExMail Encryption

SecExMail encryption uses the Twofish block cipher in conjunction with the ISAAC random number generator and is optimized to operate on real-time email streams. It uses cryptographic primitives which are available to the general public and have been subject to extensive peer review. SecExMail encryption incorporates RSA public key encryption. Message encryption is performed via the Twofish block cipher and the ISAAC random number generator. SecExMail is warranted to be free from spy-ware, key escrow or key recovery features of any kind. The email encryption process is described in detail below. See diagram.



## SecExMail Encryption



- **Stage 1**

Email data is received in variable length data blocks. SecExMail parses SMTP header info, mail and data bodies.

- **Stage 2**

Because email messages frequently contain known plain text, such as salutation and or tag lines, which gives rise to [known plain text attacks](#)<sup>[48]</sup> on the encrypted message and in order to minimize overall message expansion, the plain text is first compressed using the ZLIB compression algorithm. The net effect of deflating large amounts of data, containing both tidbits of known plain text such as greeting or tag lines as well as unknown message text into a compressed data stream is that any known plain text is effectively obscured.

- **Stage 3**

The ZLIB stream has a fixed header format which in itself might be exploited as known plain text by a savvy cryptanalyst. For this reason, the first 64 bits of the steam are encoded by way of a [One Time Pad](#)<sup>[46]</sup>, using standard XOR masking. This approach acknowledges that email messages will contain portions of known plain text and proactively manages this problem.

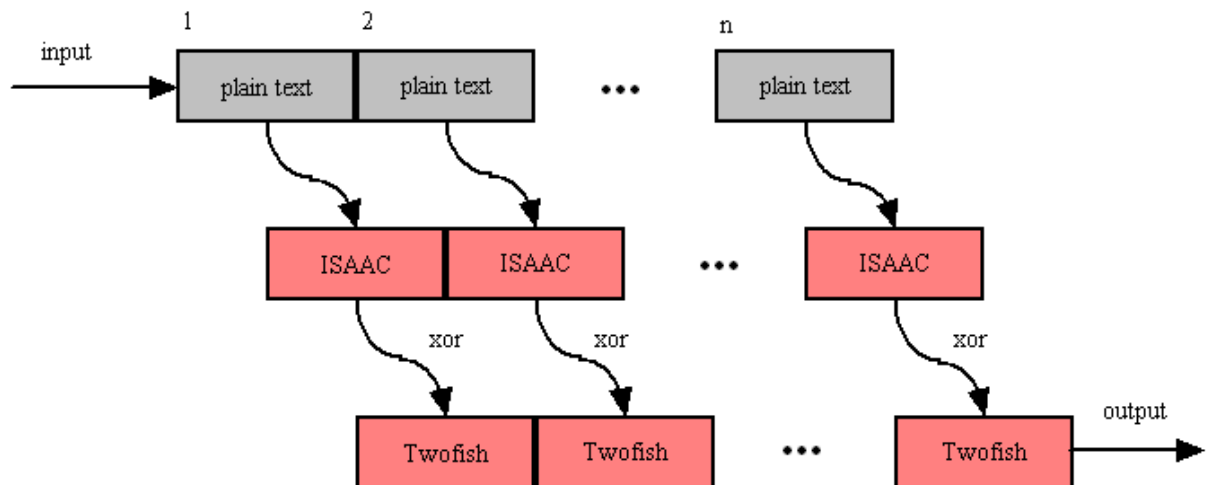
- **Stage 4**

At this point the compressed data is XOR'ed using the 64 bit ISAAC random number stream.

- **Stage 5**

The next step in the encryption process is to encrypt the random number encoded text using the 256 bit Twofish block cipher. Twofish is used in chained block mode. Instead of XOR'ing the previous block's cipher text into the plain text of the current block, the output from the ISAAC layer is "chained in". This chaining process is illustrated below.

### Twofish Block Chaining



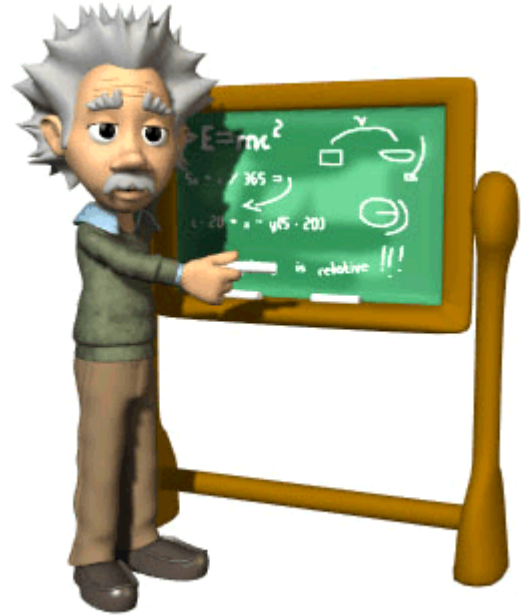
- **Stage 6**

The final step is to assemble the output in base64 transfer encoded format for transmission via mail transfer agents (MTA).

## 5.3 ISAAC Random Number Generator

ISAAC (Indirection, Shift, Accumulate, Add, and Count) is a cryptographically secure pseudo random number generator. With an average cycle length of 2 to the 8295th power its output is uniformly distributed and unpredictable. ISAAC has been developed by Bob Jenkins and placed into the public domain in 1996. See [Acknowledgements](#)<sup>[50]</sup> for legal information on ISAAC.

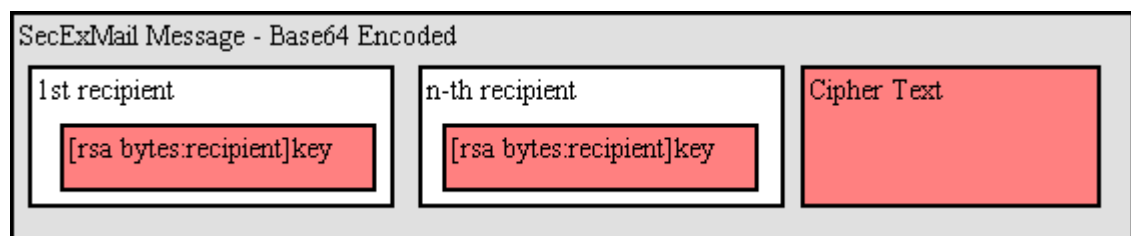
ISAAC is at the heart of SecExMail's entropy collection system.



## 5.4 SecExMail Message Format

SecExMail messages are transferred in base64 encoded format. Messages may be encrypted to multiple recipients. The internal message layout is defined as follows :

**[<rsa bytes>:<recipient>]key[<rsa bytes>:<recipient>]key...cipher text**



- **RSA Bytes**

This is the size of the recipient's RSA key in bytes. Therefore a 2048 bit RSA key would be listed as having a size of 256 bytes. RSA This parameter is defined for RSA key sizes of 2048, 4096, and 8192 bits.

- **Recipient**

This is the email address of the recipient to whom the message is encoded.

- **Key**

This is the SecExMail session key material, encrypted with the [RSA public key](#)<sup>[35]</sup> of the recipient. The SecExMail session key is used to encrypt the message body of the email message and is comprised of a 64 bit [One Time Pad](#)<sup>[46]</sup> key, a 64 bit [ISAAC random number generator](#)<sup>[38]</sup> key, and a 256 bit Twofish key.

- **Cipher Text**

This is the message body encrypted using [SecExMail Encryption](#)<sup>[35]</sup>.

A typical SecExMail message is depicted below :

```
--Begin SecEx 1.1--
WzI1NjpaHJpc0BvZmZzaG9yZWlhaWxyb29tLmNvbV0dJyyJnwwCm0LI0659zpBY/asERA3FRG9
9
OYRhm5f+rwohYORt8Wp3rmwI2Nguhk38KvH5pg8ZRTXXWiEHYMaKQPPXpbnaJepJFZeXTcNMTi/
d
p0Rc5HCTui5okW/00Gv8Sp328Ldh3D1gQcGW7oYt9qxG/cJ/PaVxxxEfDM3I4cnsCyLjfx+I0JY
6
h+emWt4U/N6u+K0tPL4ua2OfGhGoBXo+6KK042bXGpk/Pj6WEOQMCKyR+VrsOx6ZcTgpqS3WCcU
c
2/JDy9zHqkPLohXcT4G2Hiwp/1JhviaQtoKA2NYYimuY5ZjNUGPMsIaN0h6AKS3/qZsHhK1Ltc
A
WpLnuoFbQleekuJngBCC1RI1I1I4lfFgMkxoUkZrtXg6E217Q6GMMhHMANJ4EU3D2c1BgauDYAQ
G
Rpz0p8efm/WAZoXai6KVElMEiK7tv98s8wu9LpUxN44QYj2eNRVI+721GPfkBoKvr6eK5/TU4cH
N
Dg9VxCGj4n8KDvfYsPRpBSNzLL+Ta4iz7toQ/MGdPCQa
--End SecEx Mail--
```

## 5.5 SecExMail Keys

SecExMail employs public key encryption. Messages are encrypted to one or more recipients using their **public keys**. Only the intended recipient can, upon receipt of the message, recover the plain text using his/her **private key**. Public key encryption differs from classical encryption because the recipient of a message does not use the same key for decryption as the sender used for encryption.

In cryptography the fictional characters "Alice" and "Bob" are often used for illustration purposes. Consider the following scenario : Alice lives in New York and Bob lives in Los Angeles. Alice wants Bob to be able to send her confidential mail. She goes to her local hardware store and purchases a dozen or so combination padlocks, sets the unlocking code on each padlock, confuses the dials again, and sends the open padlocks to Bob in Los Angeles.



Bob is now in possession of Alice's padlocks, but not the unlocking codes. When Bob wants to send Alice a confidential letter, he places the letter inside a steel box and locks it with one of Alice's padlocks. Once the padlock is snapped shut, even he himself cannot re-open the box since he is not in possession of the combination which will release the lock. Only Alice will be able to open the box and therefore read the letter once she has received Bob's parcel in the mail.

Public key encryption works much in the same manner. The **public key** may be thought of as an open, electronic padlock. You can send this electronic padlock to all your friends. Your friends may then use that padlock to secure their emails to you in an electronic box. This electronic box is the encrypted email. Upon receipt of the encrypted email, you dial the secret combination which is your **private key** and retrieve the original message.

SecExMail does all this for you.

## 5.6 SecExMail Key File Format

The SecExMail keys are stored in conventional text files ending in ".pubrsa" and ".privrsa" for public keys and private keys respectively. Files are divided into an administrative segment and a data segment. The administrative segment contains information required by SecExMail for key management.

### Administrative Segment

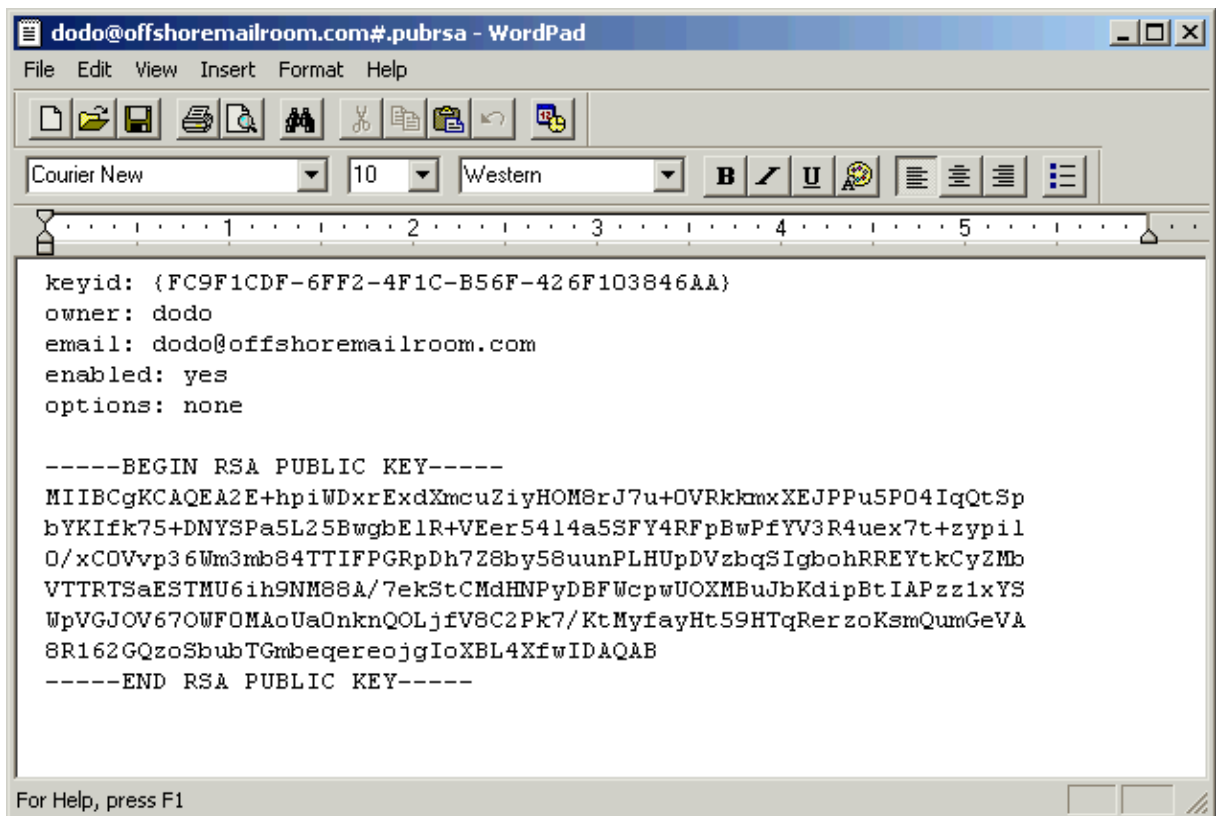


keyid	Globally unique key identifier; used by SecExMail to associate private and public key components.
owner	owner of the SecExMail key
email	Email address of key owner
enabled	reserved for future use
options	vendor options field - reserved for future use

New lines in the administrative section are denoted by carriage return line feed pairs (ASCII characters 13 + 10).

Data Segment

The data section is comprised of a single RSA key in base 64 encoded format. New lines in the data section are denoted by a single linefeed ( ASCII character 10 ). Private RSA keys are stored in 3DES encoded, chained block cipher format and protected with a passphrase.



SecExMail keys held in the registry are stored in a format analogous to keys stored on file - with each parameter represented as a registry value. See image below.

Name	Type	Data
(Default)	REG_SZ	(value not set)
email	REG_SZ	dodo@offshoremailroom.com
enabled	REG_SZ	yes
keyid	REG_SZ	{FC9F1CDF-6FF2-4F1C-B56F-426F103846AA}
options	REG_SZ	none
owner	REG_SZ	dodo
rsa	REG_SZ	-----BEGIN RSA PUBLIC KEY----- MIIBKgKCAQEA2E

## 5.7 Entropy Collection



Individual email messages are encrypted via session keys using the Twofish block cipher in conjunction with the ISAAC random number generator. Each session key is then encrypted with the SecExMail public key for the recipient of the message. Upon receipt of the message, the session key is decrypted via the recipient's private key. Once the session key for the message has been retrieved, the message itself can be decrypted. In order for the message to be secure this session key must be both random and unknowable.

Consider the following scenario where a home owner protects his garden shed with a combination pad lock. Assume further that the brand of pad lock the home owner purchased has four dials, each bearing the digits "1" through "9", and that the dials have a slight tendency to snag on the number "7". If the tendency is slight enough so as to be hardly noticeable many a buyer will, without being aware of this, chose a combination involving one or more sevens. Ordinarily a thief would be compelled to try 10,000 settings in an exhaustive search for the correct combination. On average, therefore, a thief will succeed after 5000 tries. The educated thief, however, knows that all locks of this brand have a tendency to snag on the number "7". If the thief establishes that the first two dials are so affected, then only the second pair of dials is truly unknown. With some luck, the thief only needs to examine 100 combinations, 00..99 on the second pair of dials, in order to open the lock. Our number lock, although it provides for a "key space" of 10,000 combinations has a statistical bias - some combinations are more likely than others. In order for the combination lock to be useful, its combination must be entirely unknowable.

Much the same applies to encryption keys - size does matter. But for a large encryption key to be strong, it must be unknowable to a potential attacker. This requires the input of good random numbers

during key generation. If the inputs to the key generation are not random, an attacker will be able to exploit the statistical bias. Why cut the lock, when you can simply dial the correct number ? Good randomness, unfortunately, is difficult to produce for modern computers. Computers are calculating machines which perform predefined operations according to predefined scripts, called programs. Nothing about a computer is random. Computing is 100% deterministic albeit complex and sometimes opaque to the human observer. To compensate for this shortcoming, random number generators accept what is referred to as a seed. The seed initializes the internal state of the random number generator and thus sets a starting point. Thereafter a complex mathematical sequence is applied to produce statistically pattern free output. If the starting point, or seed, to the mathematical sequence is unknowable, the random number generator can be said to be "truly random". This unknowable starting point is referred to as "entropy". The entropy of a system is the measure of its unpredictability.

Because computers are inherently deterministic, the best source of unpredictability is the human user. Many encryption systems make the mistake to digest state information about the computer, such as screen shots or process lists, gathered in short term observations into entropy data. Many encryption tools are confined to gathering entropy in this manner by the nature of their design. An encryption plugin for an email client, for example, is only invoked for a very short period of time when it is asked to encrypt an email message. It is then free to digest short term state information about the computer into entropy. The problem with this approach is that the next invocation will possibly produce similar state information. Thus if little has changed in the computer's state since the last invocation, the entropy collected at each invocation will exhibit a high degree of correlation. Some designs safeguard against this by writing a seed file to disk which transfers state information from one invocation to the next. However, the amount of entropy gathered by a program which only exists in computer memory for but a brief time is inherently limited.

For this reason, and to mine the entropy which may be found in the interaction between the human user and the computer, SecExMail operates an entropy collection subsystem which runs continuously during the operation of the computer, even when no email is being sent or received. The entropy collection extracts unknowable user data and re-seeds small subsections of the random number generator's state array as entropy data becomes available. A perfect source of randomness are keyboard timings and mouse clicks, mouse movements and mouse timings. The entropy collection subsystem **NEVER** records your keystrokes or any other user information, but exploits the timing information contained in system events generated by the user. Since some users tend to be slower typists than others, and yet other users make predominant use of the mouse, SecExMail uses two strategies to distill "unknowability" from the data it collects.

### 1) Modulus Calculation

The modulus operation is a mathematical calculation which computes the remainder of integer division. For example 7 MOD 3 equals 1. Perhaps you recall this kind of arithmetic from "Kindergarten Math". "How often does 3 go into 7 ?" Answer 2, remainder 1. This kind of arithmetic is useful in removing bias from nearly random data. For example one might conduct a survey recording the height of shoppers in a mall. When asking "random" adults how tall they are, the answers are likely to fall into a certain range : 5 foot 9, 6 foot 1, 5 foot 11, etc. While the exact answers will be unknowable to someone who did not accompany the surveyor, the collected data will not be entirely unknowable as the majority of answers will fall into a certain range. However, consider what happens when we compute the "modulus 3" of the above values.

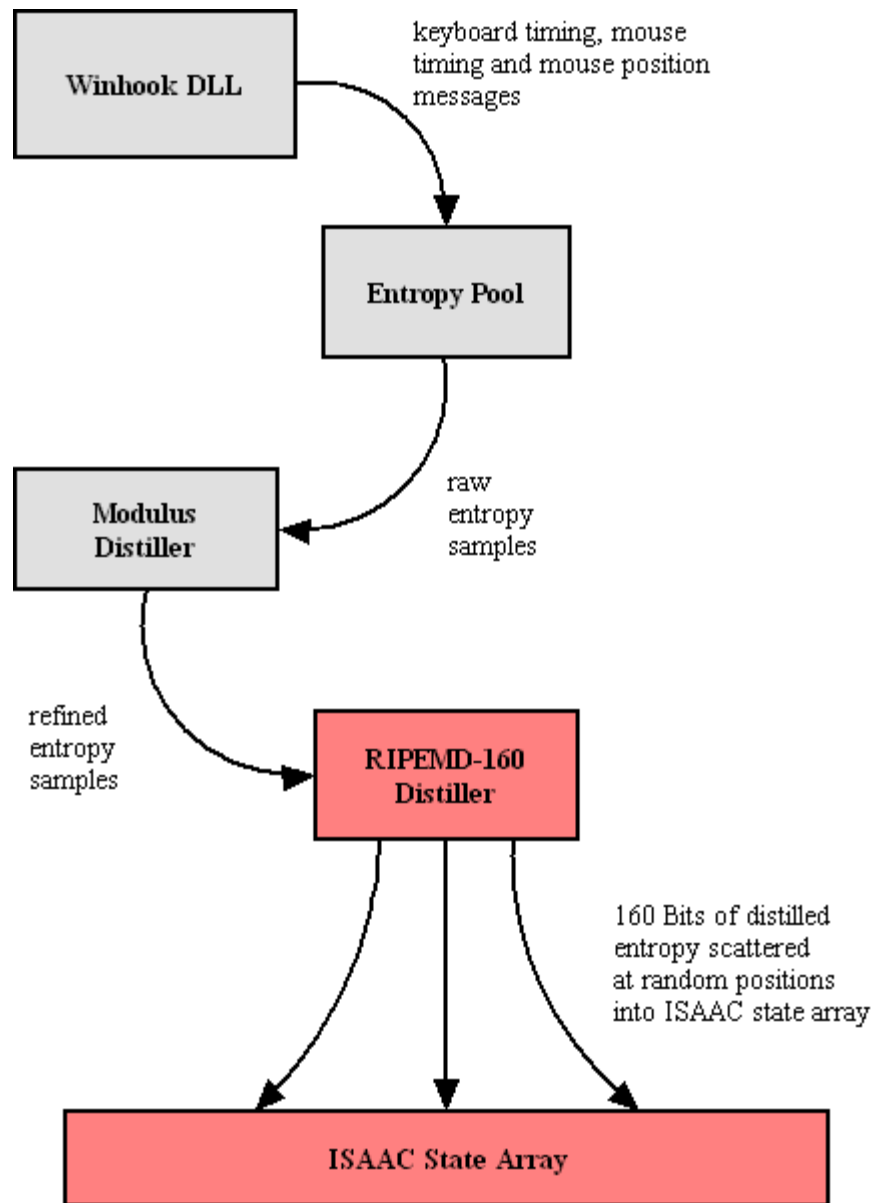
Data	Inches	Modulus 3
5 ' 9"	69 "	0 ( 23 * 3 = 69, remainder 0 )
6 ' 1 "	73 "	1 ( 24 * 3 = 72, remainder 1 )
5 ' 11"	71 "	2 ( 23 * 3 = 69, remainder 2 )

Regardless of the ethnicity of the population surveyed, someone who did not accompany the surveyor will be unable to predict the sequence of zeroes, ones and twos which will emerge. The average height of the examined population is biased, but whether a population member's height is an integral multiple of 3 inches is entirely unknowable.

## 2) Secure Hash Functions

The second strategy employed by the entropy collection subsystem to distill randomness from biased data is to apply a secure hash function. Secure, one way hash functions are used for digital signatures and cryptographic checksums. According to Ron Rivest, one of the designers of [RSA encryption](#)<sup>[35]</sup>, hash functions are designed such that "It is computationally infeasible to find two messages that hashed to the same value. No attack is more efficient than brute force." As such, hash functions tend to preserve the smallest differences in a sample and have the added property of preserving the entropy found in the sample. It is important to note that SecExMail does not use secure hash functions to "stretch" the randomness in small data sets, but to distill the entropy in data pools containing hundreds or thousands of data items to 160 bits of entropy. SecExMail employs the RIPEMD-160 message digest.

Entropy is gathered in entropy pools, distilled as described above and finally scattered randomly into the state array of the ISAAC random number generator. The diagram shown below depicts the flow of data in the entropy collection subsystem.



To prevent attacks on the random number generator based on the monitoring of key strokes and mouse events by third parties, the SecExMail entropy collection does not employ event timings which represent times at which the operating system recorded the events, but instead uses internal timestamps which represent the times at which the events were recorded by SecExMail's own message queue. This message queue is subject to further timing distortions by other events and the general multitasking behavior of the application.

## 5.8 One-Time Pads

A one-time pad is a block of random data used to encrypt a block of equal length plain text data. Encryption is usually by way of XOR'ing the one-time pad with the message text. This process may be thought of as a 100% noise source used to mask the message. The one-time pad is secure if it is comprised of random data and is never reused. Because of this, one-time pads have limited application in modern ciphers, but are commonly acknowledged as the holy grail of cryptography.

SecExMail uses one-time pads to encrypt the ZLIB compression header in [SecExMail messages](#)<sup>[35]</sup>.

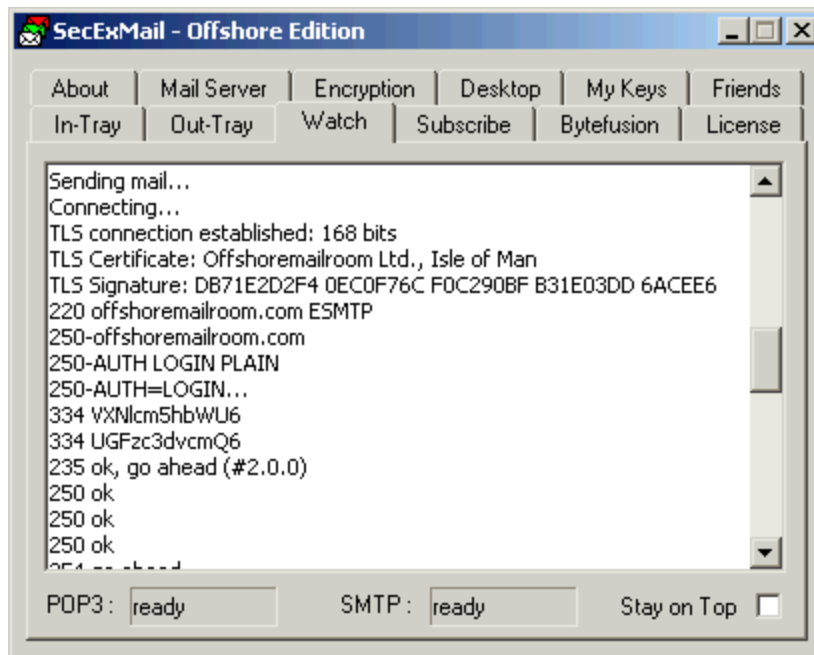


## 5.9 IP / DNS Spoofing

IP spoofing is the creation of IP packets using someone else's IP address. DNS spoofing is the substitution of a different IP address for a DNS name. DNS spoofing is commonly achieved by corrupting the DNS database of the DNS server your computer connects to in order to match human readable computer names to physical IP addresses. In both instances, the computer you are connecting to is not the server you expect.

This can be used, for example, to trick you into giving your server user name and password to the computer acting as the impostor. Alternatively, the impostor might simply act as a conduit whilst talking to the real server on your behalf. This is called a "Man-in-the-middle attack" and is commonly used to intercept network traffic without the knowledge of the original participants.

SecExMail protects against IP and DNS spoofing via SSL / TLS certificates. At the start of each connection attempt, the server certificate is verified to establish the server's true identity and the digital signature of the certificate is recorded. (Offshore and Corporate edition only)



## 5.10 Requirements

- Windows 95 / 98 / ME / NT / 2000 / XP
- SMTP and POP3 compliant email client, such as Eudora, Calypso, Outlook, etc.
- Access to internet mail server ( SMTP & POP3 )
- Pentium class IBM compatible computer



## 5.11 Known Plain Text Attack

A known plain text attack is the attempt by a cryptanalyst to break a cipher based on knowledge about the plain text of a message prior to its encryption. Simply put, if the cryptanalyst knows the method of encryption, any encryption, part or all of the plain text input to the cipher, and is able to observe the encrypted message text, he / she will likely be able to infer the key used to encrypt the message. This in turn can compromise the security of future messages sent with that key. In greatly simplified terms :

**Plain Text + Key = Cipher Text**  
**Cipher Text - Plain Text = Key**

Consider the following scenario : Alice sends Bob an email and attaches her favorite holiday snapshot. The email is encrypted. Assume further that she sends the same holiday snapshot to her mother in plain text. Steve, who wishes to spy on Alice and Bob, was able to intercept her email to Mom and now has a copy of "myholiday.jpg". If the picture consisted of 200 Kilobytes of data (about 200,000 letters) and Alice included only a short personal message to Bob with the picture ( say 50 letters ), then Steve already knows 99% of the message contents prior to encryption and now has greatly improved chances of breaking Alice's key if he comes into possession of the corresponding cipher text.

SecExMail includes comprehensive protection against known plain text attacks. See [SecExMail Encryption](#)<sup>[38]</sup> and "Proactive Security" under [What is SecExMail](#)<sup>[39]</sup>.





## 6 About

### 6.1 About SecExMail Gate



SecExMail  
Version 1.20  
Copyright © 2003, Bytefusion Ltd.  
All Rights Reserved

### 6.2 About Bytefusion Ltd.



Bytefusion Ltd.  
22 Duke Street  
Douglas, IOM  
IM1 2AY  
British Isles

Inquiries: [sales@bytefusion.com](mailto:sales@bytefusion.com)

## 6.3 Acknowledgements

- **ISAAC Random Number Generator**

At the time of writing, the ISAAC home page can be found at <http://burtleburtle.net/bob/rand/isaacafa.html>.

ISAAC has been placed into the public domain by its author, Bob Jenkins in 1996.

```
-----  
My random number generator, ISAAC.  
(c) Bob Jenkins, March 1996, Public Domain  
You may use this code in any way you wish, and it is free. No warrantee.  
-----
```

- **RSA Public Key Encryption**

The RSA algorithm was patented until September 2000 when RSA® Security Inc. released the algorithm into the public domain. *"BEDFORD, Mass., September 6, 2000 -- RSA® Security Inc. (NASDAQ: RSAS) today announced it has released the RSA public key encryption algorithm into the public domain, allowing anyone to create products that incorporate their own implementation of the algorithm."* At the time of writing a copy of this statement can be found at <http://www.rsasecurity.com/news/pr/000906-1.html>

- **Twofish Block Cipher**

The Twofish block cipher by Counterpane Labs was developed and analyzed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall and Niels Ferguson. Twofish was one of the five Advanced Encryption Standard finalists. At the time of writing the Twofish homepage can be found at <http://www.counterpane.com/twofish.html>. The cipher has been made available to the general public by the following statement on <http://www.counterpane.com/about-twofish.html> :

```
" Twofish is unpatented, and the source code is uncopyrighted and license-free; it is free for all  
uses. Everyone is welcome to download Twofish and use it in their application. There are no rules  
about use, although I would appreciate being notified of any commercial applications using the  
algorithm so that I can list them on this website. "
```

- **ZLIB Compression Library**

ZLIB is a lossless data-compression library written by Jean-loup Gailly and Mark Adler. ZLIB is made available as free, unpatented software to the general public at <http://www.gzip.org/zlib/>. The license conditions are set forth at [http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.org/zlib/zlib_license.html) and reproduced below :

```
" Copyright (C) 1995-2002 Jean-loup Gailly and Mark Adler
```

```
This software is provided 'as-is', without any express or implied  
warranty. In no event will the authors be held liable for any damages  
arising from the use of this software.
```

```
Permission is granted to anyone to use this software for any purpose,  
including commercial applications, and to alter it and redistribute it  
freely, subject to the following restrictions:
```

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org  
 Mark Adler madler@alumni.caltech.edu "

- **RIPED-160**

The RIPE message digest was written by Antoon Bosselaers for Katholieke Universiteit Leuven, Department of Electrical Engineering ESAT/COSIC, Belgium. License conditions ask us to quote the following :

```
"RIPED-160 software written by Antoon Bosselaers,
available at http://www.esat.kuleuven.ac.be/~cosicart/ps/AB-9601/ "
```

- **Viking Art - SecExMail Logo**

Katja Bengtsson of Brisbane, Australia ( katja@offshoremailroom.com )

- **OpenSSL Project**

SecExMail contains cryptographic software from the OpenSSL project at [www.openssl.org](http://www.openssl.org) which is licensed under a "BSD-style" open source licenses. These licenses asks us to state the following :

```
"This product includes software developed by the OpenSSL Project for use
in the OpenSSL Toolkit. (http://www.openssl.org/)"
```

```
"This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com). This product includes software written by Tim Hudson
(tjh@cryptsoft.com)."
```

SecExMail is an independent, derived work and no endorsement of SecExMail by the OpenSSL project is implied. The full text of the OpenSSL license and the original SSLeay License is reproduced below.

OpenSSL License

```
=====
Copyright (c) 1998-2001 The OpenSSL Project. All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in

the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:  
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org).
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:  
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====  
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

#### Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)  
All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).  
The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.

If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.

This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:  
"This product includes cryptographic software written by  
Eric Young (eay@cryptsoft.com)"  
The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).  
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:  
"This product includes software written by Tim Hudson  
(tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

- **SecExMail Encryption**

Chris Kohlhepp and Mark Robertson, Bytefusion Ltd.