Ciphire Mail
Email Encryption and Authentication

Lars Eilebrecht

Ciphire Labs
le@ciphirelabs.com

Abstract. Ciphire Mail is cryptographic software that provides email encryption and digital signatures. The Ciphire Mail client resides on the user's computer between the email client and the email server, intercepting, encrypting, decrypting, signing, and authenticating email communication. During normal operation, all operations are performed in the background, making it very easy to use even for non-technical users.

Ciphire Mail provides automated secure public-key exchange using an automated fingerprinting system. It uses cryptographic hash values to identify and validate certificates, thus enabling clients to detect malicious modification of certificates. This data is automatically circulated among clients, making it impossible to execute fraud without alerting users.

The Ciphire system is a novel concept for making public-key cryptography and key exchange usable for email communication. It is the first transparent email encryption system that allows everyone to secure their communications without a steep learning curve.

Keywords: Ciphire, secure email, email encryption, email authentication, digital signatures, certificates, fingerprints, fingerprint system, PKI.

Introduction

Ciphire Mail is cryptographic software providing email encryption and digital signatures [24]. The Ciphire Mail client resides on the user's computer between the email client (mail user agent, MUA) and the email server (mail transfer agent, MTA), intercepting, encrypting, decrypting, signing, and authenticating email communication. During normal operation, all operations are performed in the background. This makes Ciphire Mail very similar to a transparent proxy. Apart from per-user installations, Ciphire Mail may also be deployed on mail servers as a gateway solution. A combination of client and gateway installations is possible, as well.

Public-key exchange and key agreement [1] are automated and handled via certificates available through a central certificate directory. These services are operated by Ciphire Labs and do not require any local server installations, additional hardware, or additional software.

The Ciphire system provides an automated fingerprint verification system to solve trust issues existing with central certification and directory services. The Ciphire Fingerprint System allows the users of the Ciphire system to verify the authenticity of certificates and prevents them from being compromised by the provider of the central services.

Ciphire Mail uses only well-known standard cryptographic algorithms including RSA [2], DSA [3], ElGamal [4], Twofish [5], AES [6], or SHA [7] for its cryptographic operations. It uses 2048-bit keys for asymmetric algorithms and 256-bit keys for symmetric algorithms.

Installation and Integration

Ciphire Mail Client

The Ciphire Mail client consists of three parts: the core client, a graphical configuration interface, and mail connector modules (redirector). Supported email protocols include SMTP [8], POP3 [9], and IMAP4 [10]. The STARTTLS, and direct SSL [11] and TLS [12] variants of these protocols are supported as well.

For the proprietary email systems Microsoft Exchange and Lotus Notes separate connector modules are available that directly integrate with the Outlook and Notes client as a plug-in and automatically handle communication between Ciphire Mail and the email application.

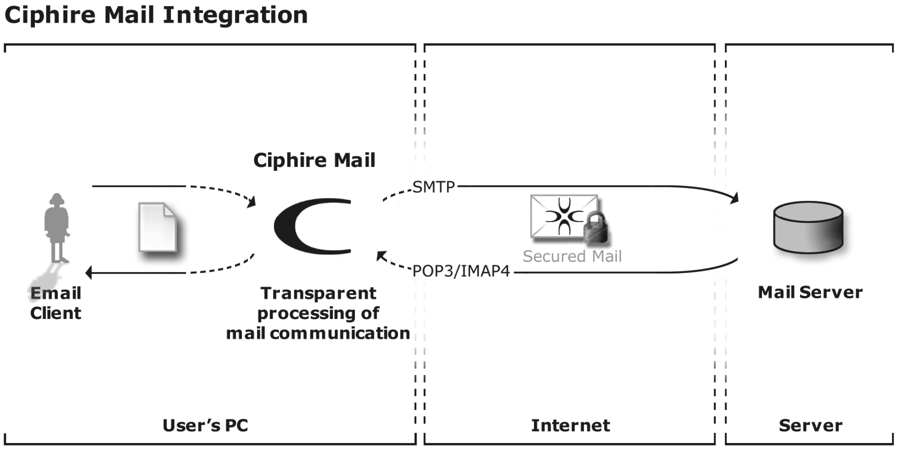


Fig. 1. Integration of Ciphire Mail

Ciphire Mail Gateway

The Ciphire Mail client can be run in “server mode” providing a gateway solution. When used in this mode, Ciphire Mail allows creation of single user certificates as well as creation of server certificates. By default, lookups are performed to find the certificate corresponding to the exact email address of the recipient. If the Ciphire Mail client or gateway finds no certificate for an email address, the lookup will automatically fall back to the domain name level.

Ciphire Certificates

Ciphire certificates use ASN.1 format [13]. This makes them similar to X.509 certificates [14], with the following exceptions and improvements.

Multiple Public Keys

Each certificate can contain an arbitrary number of public keys. Currently, Ciphire Mail uses three different keys: RSA, DSA, and ElGamal. Each certificate is signed using RSA and DSA and a Ciphire Mail client requires both signatures to be valid in order to deem the certificate as valid. Further, each message is encrypted using RSA and ElGamal (multi-layer encryption). Using always two or more different cryptographic algorithms ensures that a message or certificate will still stay secure, even if a weakness in one of the algorithms is found in the future.

Identity

A Ciphire certificate binds public keys to an email address, host or domain name. No other information about the requestor is included or required. This allows for an automated certification process.

User Controls Certification

To ensure that the user controls creation, renewal and revocation of certificate, each certificate contains self-signatures. This prevents the CA from changing or revoking a certificate without the users consent.

Revocation and Renewal of Certificates

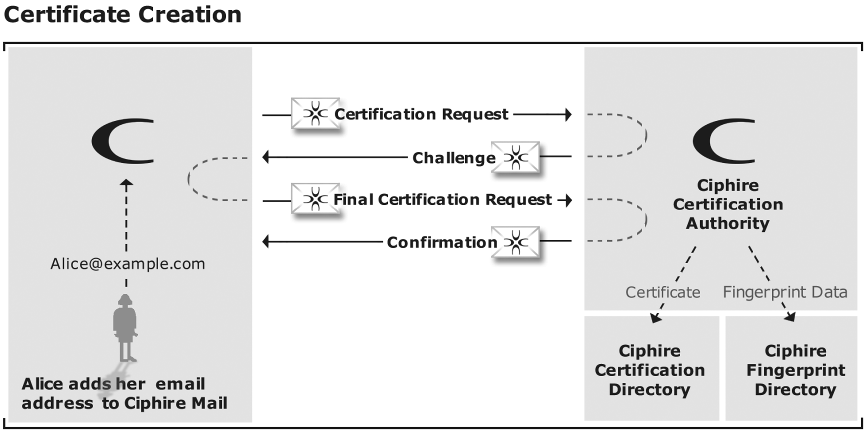
If a certificate is revoked, a dedicated revocation certificate is created. It replaces the old certificate using the same values, e.g., public keys. The renewal of a certificate involves the creation of a new set of public keys and is a combination of revocation of the old and creation of a new certificate.

Certificate Chaining

The renewal of certificate creates a cryptographic link from the old certificate to the new certificate. The revocation certificate includes the certificate ID of the new certificate and the new certificate includes the certificate ID of the revocation certificate. In addition, the revocation certificate contains a »successor signatures« created with the new keys. After a certificate has been renewed multiple times, a certificate chain is created that can be checked by the Ciphire Mail client.

Certification

Certification is an automated process invoked by a Ciphire Mail client when the user creates a certificate for a specific email address (or fully-qualified domain name). To verify the existence of the given address and to verify that the owner of the address owns the private keys corresponding to the public key, the Ciphire CA uses a mail-based challenge/response mechanism.



Certificate Creation
Certification Request
Challenge
Final Certification Request
Confirmation
Alice@example.com
Alice adds her email address to Ciphire Mail
Ciphire Certification Authority
Certificate
Fingerprint Data
Ciphire Certification Directory
Ciphire Fingerprint Directory

Fig. 2. Automatic processing of certification requests.

If all criteria for a particular certification request have been met, the Ciphire CA issues the certificate (or revocation certificate) and publishes it in the Ciphire Certificate Directory (CCD). The CA ensures that only one active certificate is available for a specific address at any given time.

Ciphire Certificate Directory

The CCD contains all certificates issued by the Ciphire CA, including active and revoked certificates. All private keys are of course created by the client and kept on the user's computer. CCD servers are part of a central infrastructure operated by Ciphire Labs. The infrastructure provides redundant services and is distributed over multiple data centers in different locations.

Every client can download certificates from the CCD by looking them up by their email address or their unique certificate ID. Lookups by email address always retrieve the current active certificate, provided one is available for the given address. A lookup by certificate ID return either the current active certificate or a revocation certificate.

All certificate lookups are fully automated and performed by the Ciphire Mail client whenever a certificate and its associated public keys are required to process a certain email message.

All lookups are cached by the client, including negative lookups that do not return a certificate. The default cache time is 36 hours, but users may configure their clients to cache lookup responses from only a few hours up to several weeks. If a Ciphire Mail client receives an email that is signed or authenticated with a new certificate, the cached copy of the certificate is automatically updated with the new certificate. Further, a user may force a remote lookup on a per-message base.

Secure Communication

The CCD is not accessed directly by Ciphire Mail clients. Instead, multiple front-end proxies are available that provide access to the CCD and other services, such as the software update, fingerprint, and time service. The core proxies are provided by Ciphire Labs, but third-party organizations are also running public Ciphire proxies. Further, organizations and Internet service provider can run a local Ciphire proxy to optimize bandwidth consumption if a large number of Ciphire users have to be served.

Communication with a proxy server is encrypted and all responses from the Ciphire services (e.g., CCD) are signed. Further, the signed response also includes the original lookup argument from the client. This ensures, that the client is able to authenticate the response and verify, that the response corresponds to his original lookup. Therefore, the proxy, or proxies, cannot change the content of the response, e.g., to return a wrong certificate.

Traffic Analysis?

A valid question regarding the CCD is: Can the provider of the CCD do traffic analysis, i.e., see who is communicating with whom?

In order to access a proxy the Ciphire Mail client has to log-on to the proxy that requires authentication with a valid Ciphire certificate. Therefore communication with a Ciphire proxy is not anonymous. This potentially allows traffic analysis on the CCD or Ciphire proxy. This kind of traffic analysis is always possible for the user's email or Internet Service Provider (ISP). However, the Ciphire system tries to minimize this risk using the following mechanisms:

- Encrypted Communication: First of all, to prevent that an external observer is able to do traffic analysis based on certificate lookups, all communication with a proxy is encrypted.
- Lookup Cache: As describe above, every Ciphire client uses a lookup cache. If a cached copy is available, the Ciphire client will not send a lookup to the proxy, until the cached response expires.
- Hashed Lookup Arguments: Lookup arguments, such as email addresses, are not included as plaintext values in the lookup. Instead a hash is calculated and used as argument. Only if the lookup yields a certificate will the proxy be able to determine the email address or certificate information of interest. Otherwise, the proxy is not able to derive any useful information from the lookup arguments.
- Daily Logons: A client does not authenticate itself for every lookup. This is only done once a day and each client is assigned a random session token during authentication. Only the session token is used to encrypt communication with a proxy. This makes it very cumbersome for the proxy to correlate lookups to the email address of a requestor.
- Primary Certificate: The certificate (i.e., corresponding private key) used for the proxy logon is not necessarily the certificate for the account that is being used as the sender of an email message. If a user has added multiple email address to Ciphire Mail, the user can choose the account that will be used to log-on to a proxy.
- Web Proxy: If a user is concerned about his IP address being known to a Ciphire proxy, the user may use a normal web proxy to communicate with a Ciphire proxy.

To prevent that the provider of the core proxies or CCD can do any kind of traffic analysis, a user may use one or more third-party proxies, i.e., proxies operated by a different organization. A lookup from a client is only forwarded by the client, but apart from the lookup argument, it does not contain any information about the client. The proxy itself logs on to an upstream proxy or one of the core proxies.

Trusted Certification and Directory Services

In many public-key cryptography solutions the user is required to blindly trust a third-party, like a classical certification authority (CA), that the issued certificate is still valid and has not been tampered with. Other systems, like OpenPGP-based systems [15], require the user to perform manual verifications of an owner's identity and integrity of a public key to find out if it is valid or not.

In the Ciphire system a user is not required to perform manual verifications and most importantly he is not required to blindly trust the Ciphire CA.

Concept

To achieve this, the Ciphire system uses, in addition to the usual CA certification, an automated fingerprinting system that provides the following:

- Verification, if a certificate for a particular user (email address) has been issued by the CA (non-repudiation of certificate issuance)
- Verification, that a certificate has not been modified after it has been issued by the CA (proof of certificate integrity)

This is achieved by the Ciphire Fingerprint System using hash-chaining techniques [16] to create a trusted log of all certification actions the Ciphire CA has performed. It makes sure, that old entries in the log cannot be changed at a later time without invalidating newer entries.

These fingerprint data is made available to all Ciphire Mail clients and used by the clients to automatically authenticate certificates. To ensure that every client has the same fingerprint data as any other client, the most current log entry (summary hash) is exchanged with other clients. When the user sends a secure email message to another Ciphire user, the client automatically includes the summary hash in the encrypted email message. The receiving client extracts the hash and compares it with the corresponding hash in its local copy of the fingerprint data. If the hash values do not match, either the sending client or the receiving client has wrong fingerprint data. The Ciphire Mail client handles all this processing automatically.

Fingerprint Creation

Fingerprints are created in the following cases:

- Certificate creation: A single fingerprint is created when a new certificate is created and issued.
- Certificate renewal: Two fingerprints are created when a certificate is renewal (one fingerprint for the new certificate and one fingerprint for the revocation certificate).
- Certificate revocation: A single fingerprint is created when a certificate is revoked (including emergency revocation), i.e., when the revocation certificate is created and issued.

- Software Update Package creation: A single fingerprint is created when a new Software Update Package is issued.

Together with information about the creation time of a fingerprint, all generated fingerprints are collected in a fingerprint list.

Fingerprint Format

A fingerprint consists of 3 cryptographic hash values (H) and a meta data field:

- $H(AID_n)$: The hash of the certificate's address ID (i.e., the user's identity in the form of an email address or hostname)
- $H(CID_n)$: The hash of the certificate's serial ID (SID) and issuer data (this hash is also called the certificate ID or CID)
- $H(C_n)$: The hash of the certificate's complete data (C_n)
- Meta data: A 2-byte binary field that defines the type of the corresponding certificate (e.g., normal certificate or revocation certificate) and shows if it has been created during certificate creation, renewal, or revocation.

With H being a 256-bit hash function (e.g., SHA_d-256), a fingerprint has a total size of 768 bit (98 byte).

Fingerprint Lists

Fingerprints are published in fingerprint lists (FPLs) with information on the creation time of the fingerprints. A fingerprint list is signed by the Ciphire Fingerprint Authority (FPA). But directly downloading all fingerprints in a single list is not feasible for a client, as the amount of data and the bandwidth consumption would be too high. Therefore, the FPA does not create a single list containing all fingerprints, but multiple lists containing a certain part of all fingerprints. Such a list is called “Branch FPL” and belongs to a certain “branch”. All branch FPL are assigned - based on their branch number - to a section and a hash over the branch FPL's contents is added to a so-called »Section FPL«. Finally a hash over each section FPL is added to a so-called “Master FPL”.

Each branch FPL contains fingerprints for a certain time interval, the FPL creation interval. An interval is usually one hour, but it may be defined according to the number of certificates issued by the CA. Finally, there are three levels of FPLs: branch FPLs, section FPLs, and the master FPL.

Fingerprints are collected for a specific interval and for every interval a set of branch, section and master FPLs are created. These are sometimes referred to as “Interval FPLs”. The interval time is not fixed, but may be changed from time to time. Common values for the interval time are in the range of 15 minutes up to 120 minutes. For example, with an interval of 60 minutes the interval start time may be 18:00:00 and the interval end time may be 18:59:59.

All interval FPLs are cryptographically linked by a carry-over hash that provides for a continuous chain of all FPLs. In addition, all interval FPLs are connected by a “Cross FPL”.

The cross FPL is a single FPL containing hashes calculated over all master FPL hashes. With each FPL creation interval an additional entry is added to the cross FPL. The cross FPL is a chronological list of hash entries. It keeps track of all certificates ever issued. Each entry corresponds to a time interval, hence to a set of interval FPLs. The main purpose of the cross FPL is to have a global hash that is known to all clients and can be verified by all clients.

This cross FPL hash and the corresponding time stamp are included in each message sent by a Ciphire client to other Ciphire clients. This functionality is called "Cross-Client Verification". With this functionality the system ensures that every client has the same FPL data. If not, i.e., if fingerprint verification fails, the user is prominently informed about the mismatch of the fingerprint entry. For example, if a user has fake or wrong FPL data, every Ciphire-secured email the user receives is going to trigger a pop-up informing the user about the security issue.

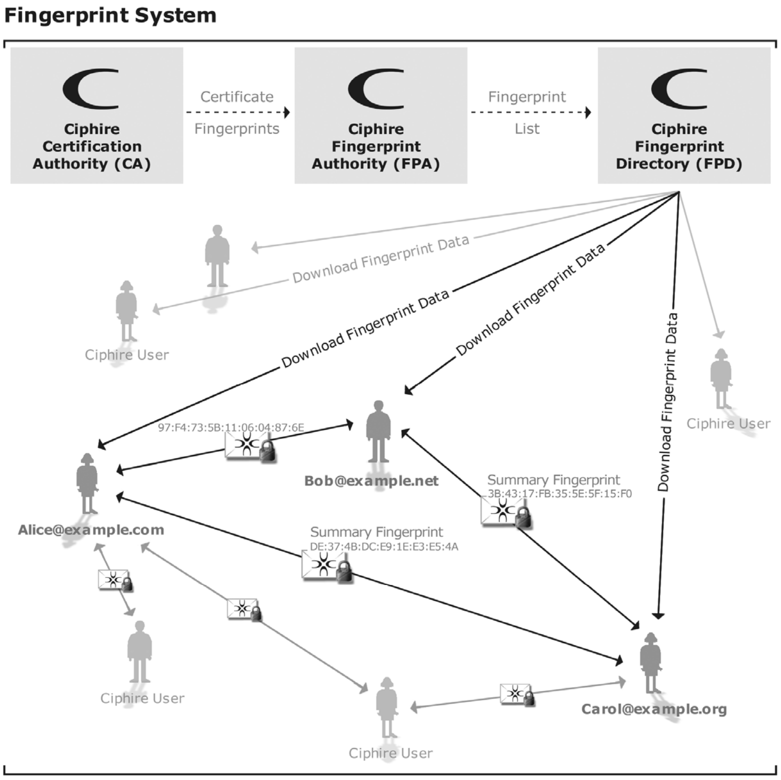

Fingerprint System
Ciphire Certification Authority (CA)
Certificate Fingerprints
Ciphire Fingerprint Authority (FPA)
Fingerprint List
Ciphire Fingerprint Directory (FPD)
Download Fingerprint Data
Download Fingerprint Data
Download Fingerprint Data
Download Fingerprint Data
Ciphire User
Ciphire User
97:F4:73:5B:11:06:04:87:6E
Bob@example.net
Summary Fingerprint
3B:43:17:FB:35:5E:5F:15:F0
Alice@example.com
Summary Fingerprint
DE:37:4B:DC:E9:1E:E3:E5:4A
Ciphire User
Carol@example.org
Ciphire User

Fig. 3. Flow of fingerprint data in the Ciphire system.

Secure Email Communication

When an email client submits a message, the redirector (mail connector module) intercepts the communication and looks up certificates for all recipient email addresses. If no certificate exists for a recipient, the client either sends the email unencrypted, rejects the email, or asks the user what to do, depending on the user's configuration.

If a lookup for an email address in the CCD yields a certificate, the client automatically downloads and validates it by verifying the certificates built-in security properties (e.g., self-signature and issuer signature). In addition, the certificate is verified with the fingerprint system described above. When the certificate is validated, the email is encrypted and sent. All this happens on the fly while the message is being delivered to the email server.

Similar steps are followed when performing decryption, and verification of digital signatures.

Authentic Emails

Signing emails may not always be desirable. To ensure that the recipient of an email is still able to identify the sender of the email message, authentication information about the sender is includes in every Ciphire-encrypted message. When a Ciphire Mail client encrypts a message, the symmetric encryption key used for this, is signed with the sender's private key. In addition to the encryption key, further data like the sender and recipient email address, a timestamp, and protocol-specific data is included in the signature. If a message's content is not digitally signed, the recipient of the message can be sure that the email of the sender address he is seeing in his email client is the authentic email address of the sender.

Tunneling Email through Email

Ciphire Mail uses a different message format for encrypted and signed emails. When encrypting an email, the whole email, including its header, is wrapped into a new email. The new email contains only minimal headers required to deliver the message. Information from the original email Subject or CC headers is only part of the encrypted contents that are put in base64-encoded form into the body of the email. The original email is tunneled through email and restored by the recipient's Ciphire Mail client. Email headers that have been added to the email while it was in transit, such as Received headers, are merged into the original email. To ensure the security of the original email, headers may be added, but a header from the original email is never overwritten with a value from the unsecure header.

Some email clients, especially when using IMAP4, download only headers of new email messages, before downloading the complete message. Therefore, Ciphire Mail includes certain email headers, e.g., the Subject, in encrypted form in the header of the encrypted email message. The encrypted data is automatically decrypted to allow these email clients to display this information.

Signing Emails

There are cases where it is desirable to send cleartext-signed email message. The problem with that is, that some mail server and especially content and virus scanner tend to modify or remove certain parts of email messages, e.g., removing the HTML part of an email message. This would break a signature if the signature has been calculated over all parts of the email message. Ciphire Mail signs every MIME [17] part (i.e., attachment) of an email message individually. This ensures that the recipient of the email message can still verify the parts it receives even if a mail server, content or virus scanner removed a certain part from the email message.

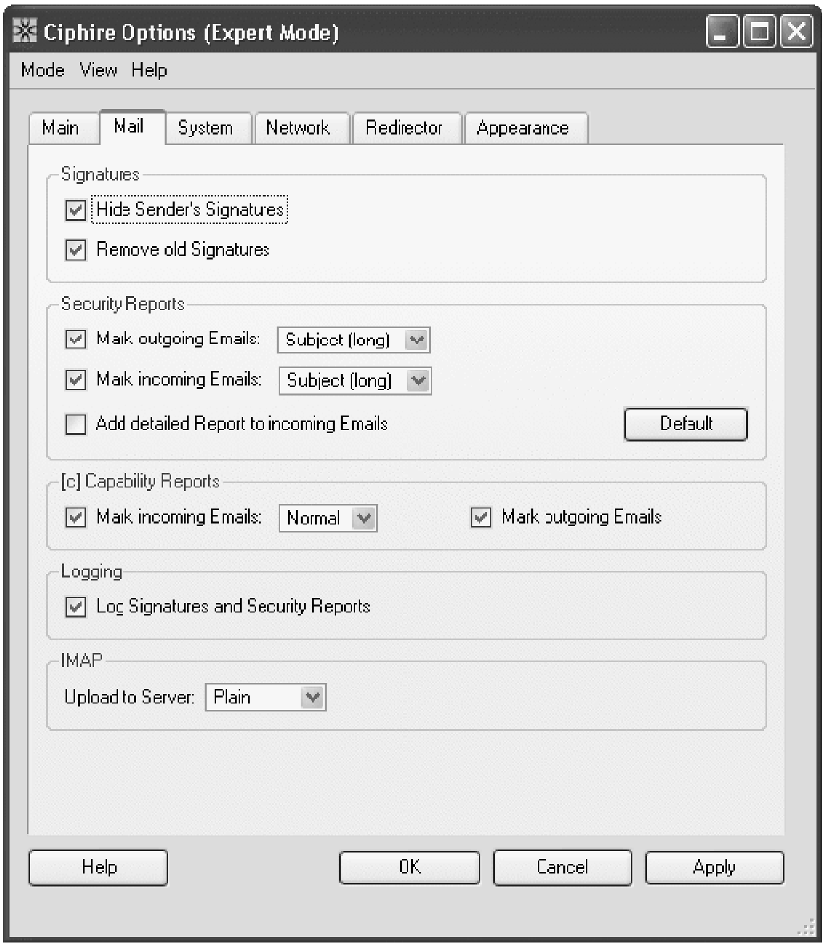
Further, a Ciphire signature always includes the email address of the sender, the email addresses of all recipients, and a time-stamp.

Status of Incoming Emails

Ciphire Mail works almost transparently, but of course it has to show the user the status of incoming email messages, i.e., if they have been received plain text or if the have been encrypted, signed, or both. This is done by putting security reports into the Subject or, optionally, From header. The user can choose between short and long reports.

- `[ciphired]` or `[es]` indicates, that the message was received encrypted and signed.
- `[encrypted]` or `[e]` indicates, that the message was received encrypted, but not signed.
- `[signed]` or `[s]` indicates, that the message was received signed, but unencrypted.
- `[u]` indicates, that the message was unencrypted and unsigned.

In addition to these reports, the user can configure Ciphire Mail to add detailed inline reports to each message.


Ciphire Options (Expert Mode)
Mode View Help
Main Mail System Network Redirector Appearance
Default
Help
OK
Cancel
Apply

Signatures

☑ Hide Sender's Signatures

☑ Remove old Signatures

Security Reports

☑ Mark outgoing Emails: Subject (long)

☑ Mark incoming Emails: Subject (long)

□ Add detailed Report to incoming Emails

[c] Capability Reports

☑ Mark incoming Emails: Normal ☑ Mark outgoing Emails

Logging

☑ Log Signatures and Security Reports

IMAP

Upload to Server: Plain

Fig. 4. Mail tab of Ciphire Mail options window (expert mode)

Controlling Outgoing Emails

Outgoing email is processed based on the user's configuration. By default all emails are encrypted if an active certificate could be found for the recipient and is automatically signed. The user can configure these settings, e.g., configure Ciphire Mail to warn the user if a message cannot be encrypted, or to not sign all outgoing emails by default. These default security strategy settings can be defined for individual recipient, e.g., for an email address, host or domain name.

However, in some cases it may be desirable to define these setting on a per-message base. This is done by putting short tags at into the Subject of outgoing email messages. Ciphire Mail checks outgoing emails for these tags and performs the appropriate action, and removes the tag from the Subject.

- `s!` - sign message
- `n!` - do not sign message
- `e!` - encrypt message (reject message, if encryption is not possible)
- `u!` - do not encrypt message
- `f!` - override local lookup cache

These tags can be combined, e.g., using `un!` would result in an unencrypted and unsigned message being sent.

Single-Point-Of-Failure?

The CCD, CA, and related services are provided as highly-available services hosted at multiple locations. Should the CCD still not be available while a Ciphire Mail client is trying to download a certificate, the user is informed about the issue and is asked if he would like to send the message in unencrypted form.

Syncronized Date and Time

The Ciphire Mail client synchronizes its internal time with the Ciphire server, i.e., the Ciphire Time-Stamping Authority (TSA). The Ciphire TSA uses the UTC time zone.

A correct time setting is important to ensure that replay attacks are not possible (e.g., when communicating with a proxy) and that signatures and certification requests from clients contain proper date and time values.

Application Requirements

Supported operating systems are Windows XP and 2000 (Service Pack 3 or higher), Mac OS X 10.3 (Panther), Linux (Kernel 2.4.0 or higher).

Ciphire Mail supports all email applications using standard SMTP for sending and POP3 or IMAP4 for receiving email (including SSL variants and STARTLS support). Microsoft Exchange and Lotus Notes will be supported in future versions of Ciphire Mail.

Cryptographic Specifications

Algorithms used in Ciphire-specific cryptographic functions:

- Asymmetric algorithms: RSA, ElGamal, and DSA-2k (DSA-2k is a variation of the normal DSA/DSS algorithm supporting 2048-bit keys [23])
- Key agreement algorithms: (not required)
- Symmetric algorithms: AES, Twofish, and Serpent [21]

- Operation modes and authentication algorithms: CBC-HMAC [19], CCM [20], and CTR
- Hash algorithms: SHA_d-256 and Whirlpool_d-512 [22]
- Pseudo-random number generation algorithm: Fortuna [18] using Twofish in CTR mode
- Supported signing modes: SHA_d-256 with DSA-2k, SHA_d-256 with RSA, and Whirlpool_d-512 with RSA

In addition to this, Ciphire Mail supports SSL and TLS and its associated algorithms. SSL/TLS is not used for Ciphire-specific cryptographic functions, but for supporting mail clients that use SSL/TLS for mail server connections. In such cases, Ciphire Mail proxies SSL/TLS connection between the email client and email server.

Availability

The Ciphire Mail tool and further information is available on the web site `www.ciphire.com`. Ciphire Mail is free of charge to home-users, non-profit organizations, and the press.

About Lars Eilebrecht

Lars Eilebrecht is Senior Security Officer at Ciphire Labs, a cryptographic research and development facility. Lars is involved in the design and development of the Ciphire system. He has a degree in computer engineering and has close to 10 years of experience in the field of Internet and security technology. In addition, Lars is co-founder and member of the Apache Software Foundation (ASF). He is a member of various ASF committees including the ASF security team and has written various books about the Apache web server.

About Ciphire Labs

Ciphire Labs is a cryptographic research and development facility with offices in Munich, Germany, and Zurich, Switzerland. The company is privately held and produces user-friendly solutions, including Ciphire Mail that enables secure communication over the Internet.

References

1. W. Diffie, M. E. Hellmann: “New Directions in Cryptography”, IEEE Transactions on Information Theory, 1976.

2. B. Schneier, et al: “Twofish: A 128-bit Block Cipher”, http://www.schneier.com/paper-twofish-paper.pdf, June 1998.
3. B. Kaliski; “PKCS #1: RSA Cryptography Specifications Version 2.0”, RFC 2437, March 1998.
4. NIST: “Digital Signature Standard (DSS)”, FIPS 186-2, January 2000.
5. T. ElGamal: “A public-key cryptosystem and a signature scheme based on discrete logarithms.”, IEEE Transactions on Information Theory, IT-31: 469-472, 1985.
6. NIST: “Advanced Encryption Standard (AES)”, FIPS-192, November 2001.
7. NIST: “Specifications for the Secure Hash Standard”, FIPS 180-2, August 2002.
8. J. Klensin, et al: “Simple Mail Transfer Protocol”, RFC 2821, April 2001.
9. J. Myers, M. Rose: “Post Office Protocol - Version 3”, RFC 1939, May 1996.
10. M. Crispin: “Internet Message Access Protocol - Version 4rev1”, RFC 2060, December 1996.
11. A. Frier, P. Karlton, P. Kocher, “The SSL 3.0 Protocol”, Netscape Communications Corp., November1996.
12. T. Dierks, C. Allen: “The TLS Protocol Version 1.0”, RFC 2246, January 1999.
13. ITU: “Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation”, ITU-T Recommendation X.680 (ISO/IEC 8824-1:2002), 2002.
14. R. Housley, et al: “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)”, RFC 3280, April 2002.
15. J. Callas, et al: “OpenPGP Message Format”, RFC 2440, November 1998.
16. S. Haber, W. S. Stornetta: “How to Time-Stamp a Digital Document”, Journal of Cryptography, 1991.
17. N. Freed: “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, RFC 2045, November 1996.
18. N. Ferguson, B. Schneier: “Practical Cryptography”, Wiley, 2003.
19. H. Krawczyk, et al: “HMAC: Keyed-Hashing for Message Authentication”, RFC 2104, February 1997.
20. D. Whitting, R. Housley, N. Ferguson: “Counter with CBC-MAC (CCM)”, http://www.macfergus.com/pub/ccm.html, 2002.
21. R. Anderson, E. Biham, L. Knudson: “The Case for Serpent”, March 2000. http://www.cl.cam.ac.uk/~rja14/serpent.html
22. V. Rijmen, P. S. L. M. Barreto: “The Whirlpool Hash Function”, http://planeta.terra.com.br/informatica/paulobarreto/WhirlpoolPage.html, 2001.
23. L. Eilebrecht: “Ciphire - Technical Product Description”, Ciphire Labs, unpublished.
24. R. Housley, N. Ferguson: “Security Design Review of the Ciphire System”, July 2004