Ciphire Mail

Next Generation Email Encryption, Now

Lars Eilebrecht

le@ciphirelabs.com

What is Ciphire Mail?

- Ciphire Mail is an email encryption tool for Windows, Linux, and Mac
- 3 years of development
- Ciphire Mail is still beta and will be released to the general public Q1/2005
- Ciphire Mail will be freely available to consumers and non-commercial organizations
- Source code will be made available as well

Who is doing it?

- Ciphire Labs
- Offices in Munich, Germany, and Zurich, Switzerland
- R&D center is in Munich
- Staffed with an international team of about 30 people

Why a new encryption technology?

- Sending encrypted messages from A to B is not a big deal ...
- ... but determining the authenticity of the public key of any given user is.
- Existing solutions:
  - Web of Trust (e.g., OpenPGP)
  - Trusted Third Party (e.g., X.509/PKIX)

Trusted Third Party?

- An X.509 certification authority (CA) certifies public keys, i.e., issues certificates
- User can verify CA signature on certificate
- Unfortunately the user has to blindly trust the CA, which is a bad thing
- CA can easily do man-in-the-middle attacks, change, or revoke certificates

Web of Trust?

- With OpenPGP, other users are required to certify other public keys by signing them.
- But a new key often has no trust path
- ... and doing a fingerprint check is a very big overhead.
- Normal users don't understand why a fingerprint check is required
- The web of trust does not work for normal users

The Goal of Ciphire Mail

- Create an email encryption tool that ...
  - ... is so easy to use, that **any** user can use it.
  - ... is so secure, that even the most paranoid security expert would use it.

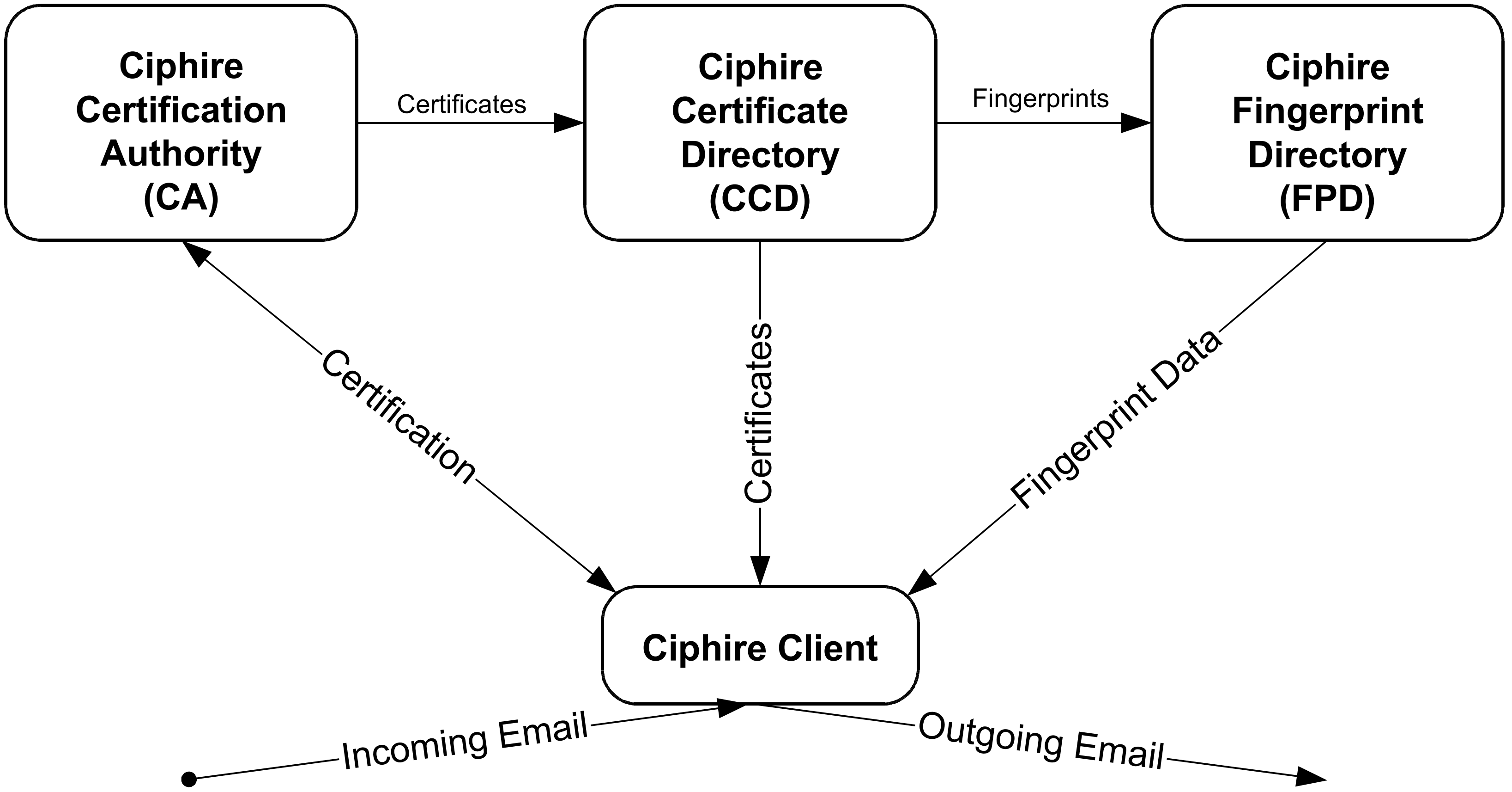
The Ciphire Mail Client

- Ciphire Mail works as a transparent proxy between the email client and the email server
- No direct integration with mail client required
- Supported standard protocols:
SMTP, POP3, and IMAP4 (incl. SSL/TLS)
- Future support for proprietary protocols:
Microsoft Exchange and Lotus Notes
- Supported Operating Systems:
Windows 2000/XP, Linux, and Mac OS X

The Ciphire System

- Ciphire Mail requires access to central services (e.g., certification and directory services)
- Interaction with the servers, such as creation and download of certificates is automated and handled by the Ciphire Mail client
- Communication with the servers is encrypted
- All server responses are signed
- Ciphire Mail client caches responses
- Ciphire is not a conventional Public-Key Infrastructure (PKI)

Main System Components



Ciphire Certificate Directory

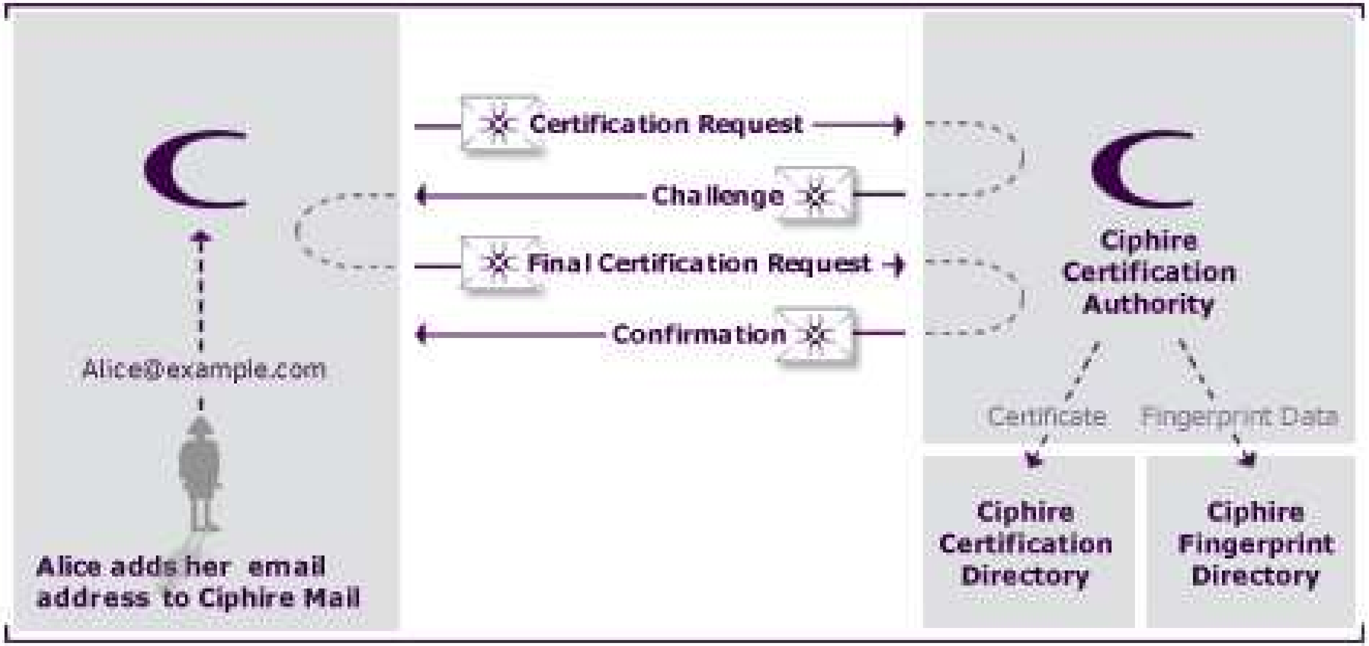
- Central directory service providing active and revoked certificates
- If a Ciphire Mail client needs a public key, the corresponding certificate is automatically retrieved
- Retrieval based on email address or a unique certificate identifier (CID)
- The Ciphire system ensures that only one certificate is active for any given identity

Certificate Identity

- Ciphire avoids the unique naming problem by using only an email address (host or domain name) as identity
- No other information about the key holder is required for the certification process
- Using only email addresses allows for an automated enrollment process

Certification Process

- Email-based certification process
- Automated and initiated by the user
- Ciphire Mail clients creates multiple key pairs
- Certification process:


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

Ciphire Certificates

- A Ciphire Certificate binds multiple public keys to an identity.
- Default: RSA, DSA, and ElGamal key
- Only one active certificate exists for a specific identity at a given time.
- A dedicated revocation certificate replaces a certificate when it is revoked.
- The private key owner controls certificate creation, renewal, and revocation.

Extended Certificate Security Properties

- Multiple certificate issuer signatures
- Multiple private key owner signatures (self-signatures)
- Certificate chain:
  - Cryptographic forward/backward links created during renewal
  - New certificate contains backward link to old (superseded) certificate
  - Old certificate contains forward link to new certificate, and a successor signature.

Securing Email Messages

- Encryption, authentication, signatures
- Email messages are processed on the fly while they are delivered to or retrieved from a mail server
- Decision if a message is to be encrypted or signed is made at the time the message is processed by Ciphire Mail, not when it is created by the user

Security Policies

- Encryption Policy:
  - Refuse unencrypted
  - Warn if unencrypted
  - Try to encrypt
  - Never encrypt
- Signing Policy:
  - Always sign
  - Never sign
- Individual Recipient Settings

Overriding Security Policies

- Subject tags can be used to override encryption/signature policy
- Signature:
  - s! - Sign message
  - n! - Don't sign message
- Encryption:
  - e! - Encrypt message
  - u! - Plain text message (don't encrypt)

Status of Received Message

- Status tags in Subject or From header

• Signed message:	[signed]	[s]
• Encrypted message:	[encrypted]	[e]
• Encrypted & signed:	[ciphired]	[es]
• Plain text message:	[u]	

- [c] in From, To, and CC header indicate if address is Ciphire-enabled

Message Format

- Complete contents is encrypted, including headers such as *Subject*
- The transmitted message contains the encrypted data in its message body
- A receiving Ciphire client decrypts the message and restores the original message
- Headers added while the message was in transit are merged into the original message

Example: Encrypted Message Structure

```
Subject: Ciphire Secured Message
X-Ciphire-Subject-1: uAIAAAAAAACAb0AAAAAQAAAQCgnM+O5oUyU1UYWf3Bc
 Zx5AAgAAcd8Baf4dpqQinAai+hcIiZFMwl/sphy/a29/WtFnziFnNP5GR4LyNUh
[...several lines of encrypted subject data...]
 WnNOKrYsItF4g1YXScEVJ8N5WclqepcEtpbrM1kQ7ORNGFqfNlQBOC+ig+3dNKP1
 Nzgc62Uo6xw=
X-Ciphire-Subject-2: [...]
From: alice@example.com

-----Begin Ciphire Message-----
uAIAAAAAAAACAb0AAAEAAQAAAQCgnM+OnFVOixoYioYQl2z5oUyU1UYWf3Bcp4N2
Zx5AAgAAcd8Baf4dpqqinAai+hcIiZFMwl/sphy/a29/WtFnziFnNP5GR4LyNUhk
[...several lines of encrypted message data...]
BvdJH+fruZn4Hj5OnzFUYOhiYIlI8pFAhj0AM7Z51TvfWbXWvJsJuAh8cnTYEBF4
ojrNbcH4efVUDFHvwenezdWoEEToLQde2cu19ZznGpnEUtOAJsoLCA==
-----End Ciphire Message-----
```

Digitally Signing a Message

- Signature includes From and To email address, and Date.
- Inline signatures for text and HTML messages
- All attachments (MIME message parts) are signed individually

Example of Ciphire Signature

```
---------------------[ Ciphire Signature ]---------------------
From: alice@example.com signed "contract.pdf" (664223 bytes)
Date: on 28 December 2004 at 14:23:42 GMT
To:   bob@example.com, carl@example.net
---------------------------------------------------------------
00fAAAAAEAAAD+WZ9ABAEAALgCAAIAAgACACAxtN4blwNOgpZbT2j9Gm84OPsAO
COTr17U+wzYy8P7QEAd64CrcECnu6qeOQRHlgGd+wrPwq99XEn/3sgO4Twmnpzu
q1wP6ioxV5kn3WLy7lMWdTx2IvlVujFeifEe18/A==
------------------[ End Ciphire Signed Message ]---------------
```

Authenticating a Message

- Any secured message (even if not signed) contains authentication information of the message originator
- The sending Ciphire Mail client signs the session key used to encrypt the message

Trust Model

- Ciphire system employs a hybrid trust model
- Hierarchical trust model elements known from conventional PKI solutions
- Distributed trust model elements specific to the Ciphire system that prevent the system from being compromised

Hierarchical Trust Model Elements

- Strict hierarchical trust model:
  - Ciphire Root CA issues Ciphire CA certificate
  - Ciphire CA (in corporation with the user) issues user certificates
  - Users sign email messages
- Authorization
  - A certificate is only deemed valid if the issuer and self-signatures are valid, and if the certification path leads to the Ciphire Root CA

Distributed Trust Model Elements

- Automated fingerprint verification system
  - Ciphire CA creates and publishes fingerprint data
  - Ciphire clients download and verify fingerprint data
  - Ciphire clients exchange and compare fingerprint data as part of normal email communication

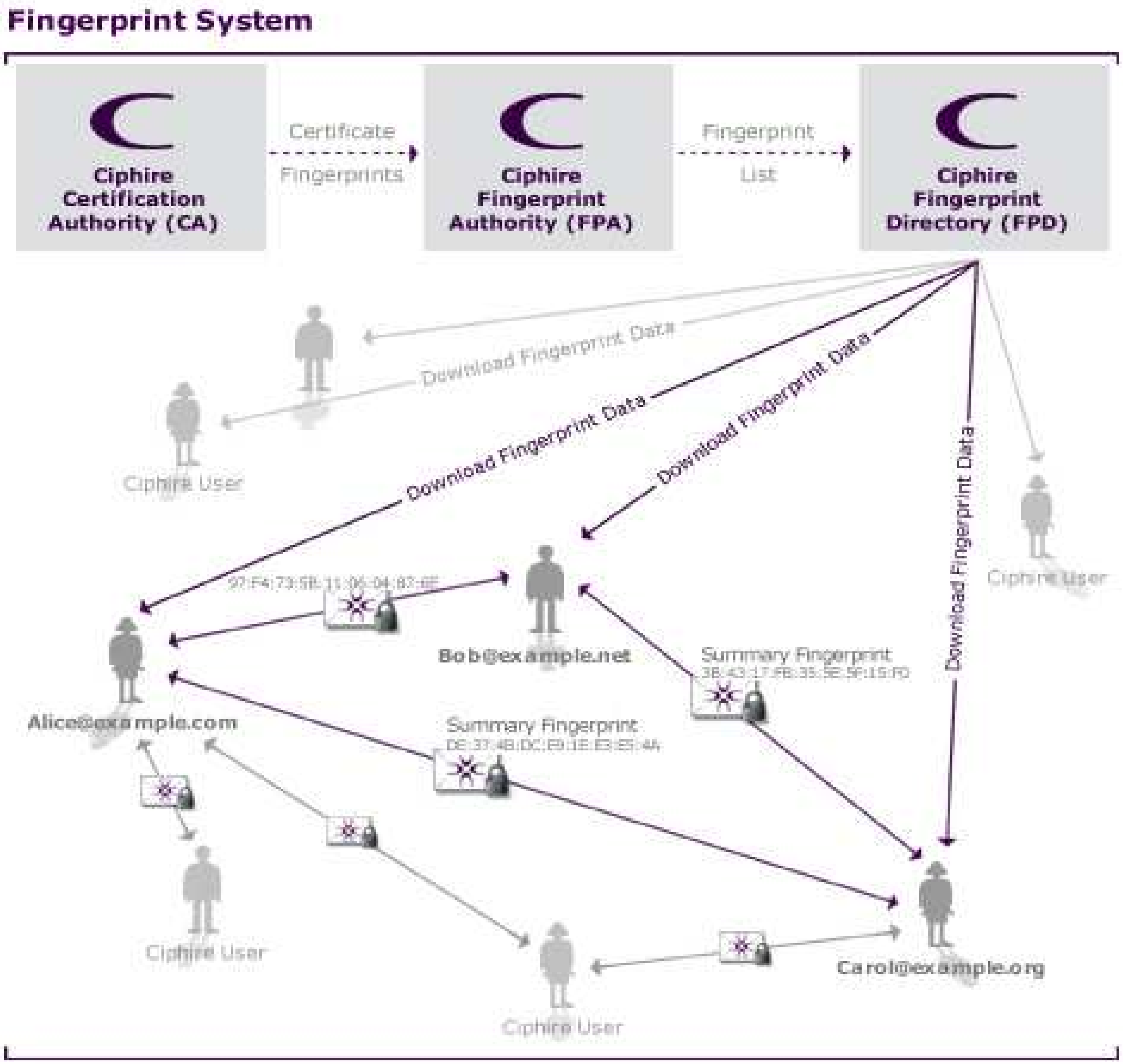
Ciphire Fingerprint System

- Concept:
  - Whenever the CA issues a new certificate or revokes an old certificate, a hash value H(C) over the certificate is calculated (the certificate fingerprint).
  - A new summary hash H(S) is calculated over the new hash and all previous hashes (hash chaining).
  - H(C) and H(S) are stored together with a time stamp as an entry in a log.
  - The log with hash/fingeprint data is made available to all Ciphire clients
- Distribution via directory service from the central Ciphire infrastructure

Cross-Client Verification

- Each encrypted email message contains a copy of the most current summary hash
- Receiving client verifies the summary hash against local copy of fingerprint data
- This ensures that all users of the system have identical fingerprint data

Ciphire Fingerprint System


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:FD
Alice@example.com
Summary Fingerprint
DE:37:4B:DC:E9:1E:E3:E5:4A
Ciphire User
Carol@example.org
Ciphire User

FP System: Implementation Details

- Not a single log of fingerprints
- Data is split in smaller parts, forming a tree-like structure
- Provides performance and scalability for an arbitrary number of users
- Single fingerprint contains multiple hash values
    - address ID, certificate ID, and certificate data

Fingerprint Format

- A single fingerprint consists of the following values:
  - H(AID): The hash of the certificate's address ID
  - H(CID): The hash of the certificate ID (serial no.)
  - H(C): The hash of the entire certificate
  - M: A 2-byte meta-data field
    (defines certificate creation, renewal, or revocation)
- SHA_d-256 is used to compute each of these values, resulting in a total size of 98 byte per fingerprint.

Fingerprint Lists (FPLs)

- Fingerprint data is partitioned into multiple lists of fingerprints for particular time intervals

- Inteval FPLs: Set of lists for each interval
- Cross FPL: Top-level summary list
  - Cross FPL hashes are distributed between client in their normal email communication

Interval and Cross FPL Structure

- Interval FPLs
  - Branch FPLs: leaves of the tree, containing certificate fingerprints
  - Section FPLs: summary hashes of branch FPLs
  - Master FPL: root of the tree with summary hashes of section FPLs
- Cross FPL
  - Summary hashes from each Master FPL
  - Entry consits of hash and time-stamp
  - Lists grows over time

Protection

- The system protects against the following (intentional or unintentional) malicious actions which could be performed by the Ciphire CA or related authority systems:
  - Malicious replacement of the public keys in a certificate (e.g., to allow man-in-the-middle attacks)
  - Malicious changes to one or more certificate fields, such as the validity dates or email address in the subject of the certificate

Trust Model Summary

- Security concept:
  - each client checks its own certificates against the fingerprint data
  - each client checks other certificates against the fingerprint data
  - each client compares summary hash with each communication partner
- This makes it impossible to perform malicious actions, without alerting many users that something is wrong.

Time-Stamping Service

- Ciphire clients syncronize time with the Ciphire Time-Stamping Authority (TSA)
- TSA uses the UTC time zone (GMT +0)
- Time is kept internal to the Ciphire client

- A correct time setting is important to ensure:
  - that replay attacks are not possible (e.g., when communicating with a proxy),
  - that a signature contains a proper time stamp,
  - that a CSR contains proper userValidity values.

Cryptographic Functions

- Ciphire certificates and software are not limited to specific algorithms or specific key sizes.
- For additional robustness the Ciphire system uses two or more different cryptographic algorithms for encryption functions.
- Even if one algorithm is broken, the Ciphire system will still be secure.

Asymmetric Algorithms

- RSA for digital signatures and encryption
- ElGamal for encryption
- DSA-2k for digital signatures
  - Ciphire uses the DSA algorithm with a 2048-bit prime and a 256-bit group order
- Key sizes: 2048 bit
- Internal tests have shown that using 16+ kbit keys are not fun anymore.

Symmetric Algorithms

- AES
- Twofish
- Serpent
- Cipher block modes:
  - CBC-HMAC, CCM (Counter with CBC-MAC), CTR
- Nonce-based (random lead-ins)
  - Protection against chosen plain text attacks
- Key size: 256 bit

Hash Algorithms

- SHA_d-256
- Whirlpool_d-512

- The “d” denotes double-hashing mode which eliminates any possibility of length extension attacks

Random Number Generator

- “Fortuna” pseudo-random number generator (PRNG)
- The Ciphire implementation of Fortuna uses the Twofish cipher in counter mode with multiple OS-specific entropy sources.
- Fortuna has been published by
  Niels Ferguson and Bruce Schneier

www.ciphire.com