Certified Mailing Lists

Himanshu Khurana and
Hyung-Seok Hahm
NCSA, UIUC

Introduction: Mailing Lists (MLs)

**List Moderator (LM)**  
- creates lists  
- Subscribes users

**List Server (LS)**  
- creates lists  
- forwards emails  
- archives email

**User/subscriber**  
- subscribes to lists  
- sends/receives email

- MLs enable users to easily exchange email  
  - LS bears overhead

- Increasingly popular for exchange of sensitive content  
  - MLs of security administrators, healthcare researchers  
  - Need confidentiality, integrity, authentication

- Potential use for dissemination/exchange official documents  
  - Multi-party contract negotiations, official announcements  
  - Need **certified delivery**
Introduction: Certified Email

- **Aim**: recipient gets email *iff* sender gets receipt
  - Derive from fair exchange of digital goods
  - Close in nature to fair non-repudiation

- **Two categories**
  - *Inline*: active TTP ensures fairness
  - *Optimistic*: rely on TTP for dispute resolution

- **Large body of existing work**
  - Two party, multi-party

- **Unique requirements for Certified MLs (CMLs)**
  - Must retain ease of offloading management to LS
  - Existing work burdens sender with key and receipt management
Overview/Highlights of Proposed Solution

- **Certified Mailing List Protocol (CMLP)**
  - Provides certified delivery, confidentiality, integrity, and authentication
  - Use public key *proxy encryption* techniques for delivery, confidentiality

- **Inline protocol**
  - LS existing on-line entity
  - Need on-line entity for offloading message processing
  - Co-locate LS and TTP (L/T) for simplicity
    - not necessary for correctness

- **Formal Verification**
  - Use Proverif - fully-automated protocol verification tool
  - Previously used for verification of two-party certified email protocol [AB03]
Background: Proxy encryption*

- **Basic idea**
  - Convert ciphertext for one key into ciphertext for another key without revealing secrets keys or cleartext messages

- **Example construction based on El Gamal**
  - Keys for A, B: \((SK_A, PK_A), (SK_B, PK_B)\)
  - Proxy key for transformation agent \(T\): \(\pi = (SK_B - SK_A)\)
  - Encryption of \(m\) for Alice: \(g^r, m(PK_A)^r\)
  - Transformation of mesg for Bob by \(T\): \(g^r, m.(PK_A)^r.(g^r)\pi = g^r, m.g^{(SK_A + SK_B - SK_A)} = g^r, m(PK_B)^r\)

- **Applications**
  - Key escrow without revealing secret keys
  - Smartcard key management, personalization schemes
  - File sharing
  - Confidentiality in MLs

* Initially developed by [MO97, BGM98]; later studied by [ID03, AKGH05]
CMLP Requirements and Approach

- **Strong Fairness**: either all parties get what they want or no one does
  - Subscriber gets email $\Rightarrow$ *all* list subscribers get email *and* sender gets receipt from *every* subscriber
  - *Approach*: Inline L/T can provide strong fairness

- **Weak Fairness**: every pair of sending and receiving parties get what they want or no one does
  - Subscribers get email $\Rightarrow$ sender gets receipt
  - *Approach*: Inline L/T can provide weak fairness

- **Non-repudiation of origin/receipt**: a party cannot falsely deny originating (receiving) a message
  - *Approach*: Use digital signatures
CMLP Requirements and Approach

- **Confidentiality**: only sender/receivers can access plaintext
  - Two approaches in the past
    - Sender encrypts with receivers’ public keys (or group)
      - [BCP03, Z04, KM00]
    - Sender encrypts with TTP’s public key but TTP does not see message
      - [AGHP02]
  - Neither works for MLs
    - Sender cannot be burdened with key management
    - L/T forwards emails so sees them
  - **Approach**: Use proxy encryption with L/T transforming messages
    - [KSB05 - SEALS]

- **Authenticity and Integrity**: parties should be able to verify identities and detect message modification
  - **Approach**: Use digital signatures
Protocol Design Choices

- **Semi-trusted neutral party** (L/T)
  - Neutral party trusted for protocol completion but not with access to “goods”
  - Neutral party does not collude with users [FT98]

- **Fairness provided via control of decryption key**
  - Release decryption key and receipt simultaneously
  - Used for fair non-repudiation [KM00]

- **Integrate (not layer) fair exchange and SELS**
  - *Optimization*: sender need not send separate message for releasing decryption key
Protocol Overview: List Creation and User Subscription

Creating a List:

\[ \text{LM} \rightarrow \text{“Create” List L} \rightarrow \text{L/T} \]

\((K_{LM}, PK_{LM}) \rightarrow (K_{L/T}, PK_{L/T}; PK_{LK} = PK_{L/T} \cdot PK_{LM})\)

Subscribing Users:

\[ \text{U}_i \rightarrow \text{Request: “Join” List L} \rightarrow \text{LM} \]

\[ \text{L, TK}_{U_i}, \text{Ticket} \rightarrow \text{LM} \]

\[ \text{U}_i \leftarrow \text{LM} \]

\[ \text{L, r, Ticket} \rightarrow \text{L/T} \]

\[ \text{(computes } K'_{U_i}, \text{PK}_{U_i}) \rightarrow \text{(computes proxy key } K'_{U_i}) \]
Protocol Overview: Sending Certified Emails

1. $\text{Enc}_k(\text{Sig}_S(m)), \text{Sig}_S(h_m^*), \text{Enc}_{PK_{LK}}(k)$

2. $\text{Enc}_k(\text{Sig}_S(m)), \text{Sig}_S(h_m), \text{"new email"}$

$L/T$ waits for key requests from all receivers

3. $\text{Sig}_{R_i}(h_m), \text{"key request"}$

$L/T$ waits for key requests from all receivers

4. $\text{Sig}_{L/T} (\text{Enc}_{PK_{R_i}}(k), h_m), \text{"key delivery"}$

5. $\forall_{i=1}^{t} \text{Sig}_{L/T}(\text{Sig}_{R_i}(h_m)), \prod_{i=1}^{t} \text{Sig}_{L/T}(\text{Sig}_{R_i}(h_m)), \text{"del. rcpt"}$

$L/T \rightarrow S$

* where $h_m = H(\text{Enc}_k(\text{Sig}_S(m)))$
Discussion and Examples

- User overhead minimized; L/T bears overhead
  - User maintains a list encryption/decryption keys
    - Key not affected by join/leave operations
  - Sender need not send separate message releasing k
    - Improvement over previous schemes
  - Sender only has to verify one signature for receipts
    - Using condensed signatures [MNT04]

- Examples of CMLP Use
  - Multi-party negotiations
    - E.g., Contract signing, establishing coalition resource-sharing agreements
    - Strong fairness for messages in the negotiation and final contract signing
  - Information Dissemination
    - E.g., Information regarding new Visa regulations
    - Weak fairness suffices
Formal Verification

- **Confidentiality** provided by proxy encryption scheme
  - Reduction proof showing our scheme is CPA secure
    - or El Gamal is not

- **Weak Fairness** proven using Blanchet’s Proverif
  - Sound verification tool with formal language
  - Tool translates protocol into Horn clauses and applies resolution-based solving algorithm
  - Capability: allows for establishing correspondence assertions
    - Execution of end(M) implies execution of begin(M)
    - Use this to prove that receiver gets email iff sender gets receipt
Future Work and Challenges

- **Deployability**
  - Compatibility with existing email infrastructure
  - Additional hardware, software, administrative costs
  - Providing effective key management
    - Generate, distribute, trust and use public/private keys

- **Usability**
  - Installation and setup
  - Interfaces and metaphors

- **Future Work: Build on Secure ML prototype**
  - Integrated with GPG
  - Plugin for LS (Mailman)
  - Usability studies planned
Questions?

- Contact hkhurana@ncsa.uiuc.edu
- Paper available at: http://www.ncsa.uiuc.edu/people/hkhurana