SELS:
A Secure E-mail List Service*

Himanshu Khurana
NCSA

Work done with Adam Slagell and Rafael Bonilla

Introduction to E-mail List Services

- E-mail List Services (ELSs) comprise
  - List Moderator (LM) – user/process that creates lists and controls list membership
  - List Server (LS) – creates lists, maintains list membership information, forwards e-mails, and optionally archives them
  - User/subscribers – subscriber to/ unsubscribe from lists with the help of LM, and send/receive e-mails with the help of LS

- Increasingly popular for exchange of both public and private content ⇒ security is an important concern
  - E.g., there are over 300,000 registered lists on LISTSERV while only 20% of them serve public content
Focus: Security Solutions for ELSs

- Various security solutions for two-party e-mail exchange (TPEE)
  - E.g., solutions for message confidentiality, integrity, authentication, and anti-spamming

- However, little or no work in providing similar solutions for ELSs

- In this work
  - Argue that a different set of solutions is required for ELSs
  - Provide solutions without adding additional components and imposing minimal overhead
    - Confidentiality, integrity, authentication, and anti-spamming
  - Describe a prototype implementation
Confidentiality, Integrity, and Authentication with S/MIME for TPEE

Alice

Base-64 encoded message

Bob

<table>
<thead>
<tr>
<th>Email Plaintext $m$</th>
<th>Sign $h(m)$ with $SK_A$</th>
<th>Encrypt $(m, Sig(m))$ with $k$</th>
<th>Encrypt $k$ with $PK_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g., SHA-1 followed by RSA/DSS</td>
<td>e.g., 3DES, AES</td>
<td>e.g., El Gamal, RSA</td>
<td></td>
</tr>
</tbody>
</table>

- Other approaches for achieving confidentiality
  - PEM, PGP
    - Challenge: certificate distribution and validation
  - IBE (Boneh & Franklin), Identity-based mediated-RSA (Ding, Tsudik)
- Other approaches for achieving integrity and authentication
  - PEM, PGP
  - Challenge: certificate distribution and validation
Confidentiality, Integrity, and Authentication with PGP for TPEE

PKI: Bob’s encryption key $PK_B$

PKI: Alice’s signature verification key $PK_A$

**Alice**

**Base-64 encoded message**

<table>
<thead>
<tr>
<th>Email Plaintext $m$</th>
<th>Sign $h(m)$ with $SK_A$</th>
<th>Encrypt $(m, \text{Sig}(m))$ with $k$</th>
<th>Encrypt $k$ With $PK_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>e.g., SHA-1 followed by RSA/DSS</td>
<td>e.g., AES</td>
<td>e.g., El Gamal, RSA</td>
</tr>
</tbody>
</table>

- **El Gamal encryption with key $(g, p, PK_B)$ where $PK_B = g^{SK_B} \pmod{p}$ in $\mathbb{Z}_p^*$**
  - choose a random number $r < p$
  - compute $c_1 = g^r \pmod{p}$; $c_2 = (PK_B)^r k \pmod{p}$
  - send $(c_1, c_2)$

- **El Gamal decryption with key $(g, p, SK_B, PK_B)$**
  - compute $c_1^{SK_B} \pmod{p} = g^{rSK_B} \pmod{p} = (PK_B)^r \pmod{p}$
  - compute $k = c_2 / c_1^{SK_B} \pmod{p}$
Confidentiality, Integrity and Authentication for ELSs

- **Confidentiality**
  
  ![Diagram showing encryption and decryption process]
  
  Problem: adversary can compromise LS to obtain e-mails

- **Possible Solutions**
  
  - Use cryptographic hardware to do decryption/encryption
    - Expensive, broken in the past (e.g., CCA application on IBM 4758)
  
  - Use group key management to distribute symmetric key to list members
    - Members need to be online to execute operations for key update

- **Our solution**
  
  - Use software-based proxy re-encryption at LS to transform encrypted e-mail between sender and receivers without requiring access to plaintext
    - Inexpensive, does not require members to be online for updates
    - Integrate (encryption) key distribution with user subscription

- **Integrity and Authentication**
  
  - Authentication goal includes notion that member must be a list subscriber
  
  - Our approach: Use digital signatures with LM providing certificate validation (w.r.t. list membership)
Anti-Spamming

- **Pricing Functions** (Proof of Computation)
  - Moderately hard CPU functions
    - Use hints to reduce difficulty of hard problems; e.g., extract square roots mod p with variable \(|p|\), hashCash
  - Memory bound functions
    - Independent of processor speeds; e.g., table-based searches that guarantee a min number of cache misses
  - Challenge: Trusting authorities that moderate use of pricing functions

- **Filters**
  - Keyword based; e.g., “be over 21”
  - Naive Bayesian
    - Train filters with sample e-mail
  - Challenge: False positives
Anti-Spamming

- **Legal**
  - Spam is a felony in VA
  - In CA, PA sexually explicit emails must have pre-fix: “ADV: ADLT”
  - Congress passed CAN SPAM act effective Jan 2004
    - Appropriate labeling, opt-out options
  - Challenge: Tracing, enforcement, jurisdiction

- **Digital Signatures**
  - Reject e-mail if signature does not match
  - Challenges
    - Large scale certificate distribution/revocation
    - Block use of e-mail for initiating communication/transactions

- **ELSs Goal: only subscribed users can send e-mail to list**
  - Use digital signatures with LM providing certificate validation
  - Use MACs as a cheaper alternative with LS participating actively
Introduction to 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 $T$: $g^r, m(PK_A)^r(g^r)^\pi = g^r, m.g^{r(SK_A+SK_B- SK_A)} = g^r, m(PK_B)^r$

- **Applications**
  - Key Escrow without revealing secret keys
  - Smartcard key management, personalization schemes

- **Challenge**
  - Trusted entity generates proxy key

Contribution: SELS; Solutions for

- **Confidentiality**
  - Solution using proxy encryption techniques whereby the plaintext is not exposed at LS; instead, LS simply transforms encrypted messages
    - LS archives e-mails in encrypted form and provide access on-demand
  - User subscription process used to distribute encryption keys without the need for a trusted third party

- **Integrity and authentication**
  - Solution using digital signatures where certificate validation is provided by LM

- **Anti-spamming**
  - Solution using digital signatures and MACs where LS discards any message not sent by a valid subscriber
SELS Overview

- Assumptions
  - LM is an independent entity not controlled by LS
  - Subscription e-mails between user, LM, and LS can be secured (e.g., PGP, passwords)
1. "Join" request
2. Choose r
3. TK = K_{LM} + r, Ticket(LS, r)
4. Choose r'
5. K_{U1} = TK + r'
6. r', Ticket
7. K'_{U1} = K_{LS} - r - r'

- K_{U1} + K'_{U1} = K_{LK}
- Only U1 knows K_{U1}
- U1 and LS compute H_{U1} = h(r')
SELS (Proxy) Encryption Scheme

\[ K_{LM} + K_{LS} = K_{Ui} + K'_{Ui} = K_{LK} \]

Encrypt (El Gamal) with \( g^{K_{U1}} \)

Transform* with \( K'_{U1} \) and \( K'_{U2} \)

Decrypt (El Gamal) with \( K_{U2} \)

\[ X = g^r, (g^{K_{U1}})^rm \]

* \( \Gamma_{(K'_{U1}, K'_{U2})}(X) = g^r, m(g^{K_{U1}})^rg^{K'_{U1}}(g^{rK'_{U2}})^{-1} = g^r, (g^{K_{LK} - K_{U2}})^rm = g^r, (g^{K_{U2}})^rm \)

(We show that the SELS encryption scheme is as secure as El Gamal in the random oracle model)
**Sending E-mails**

**Key Store:** $(SK_A, PK_A), H_{UA}$

---

**Alice**

- **Base-64 encoded message**
  - Email Plaintext $m$
  - $\text{Sig}(h(m))$ with $SK_A$
    - e.g., SHA-1 followed by RSA/DSS
  - $\text{Encrypt} (m, \text{Sig}(h(m)))$ with $k$
    - e.g., 3DES, AES
  - $\text{Encrypt} k$ with $PK_A$
  - $h(X)$ with $H_{UA}$

---

**Bob**

- **Base-64 encoded message**
  - Email Plaintext $m$
  - $\text{Sig}(h(m))$ with $SK_A$
    - e.g., SHA-1 followed by RSA/DSS
  - $\text{Encrypt} (m, \text{Sig}(h(m)))$ with $k$
    - e.g., 3DES, AES
  - $\text{Transform} k$ with $K'_{UA} K'_{UB}$
  - $h(Y)$ with $H_{UB}$

---

**Key Store:** Members’ corresponding private keys $K'_{Ui}$

---

**Key Store:** $(SK_B, PK_B), H_{UB}$
Additional Protocol Operations

- **Unsubscribing Users**
  - User sends request to LM who cosigns request and sends to LS
  - LS deletes user’s corresponding private key

- **Certificate validation**
  - LM sends occasional (e.g., monthly) updates on invalid certificates

- **Archiving and retrieval**
  - Archive after partial transformation; encrypted with $K_{LK}$
  - Complete transformation for user on request

- **Analysis**
  - If users have access to all e-mails, compromise of one user leads to compromise of all e-mails

- **Solution: Use key epochs**
  - Archive e-mails only for an epoch
  - LM changes $K_{LK}$ every epoch (efficient: one e-mail from LM)
    - LM $\rightarrow$ Users: $r$ (via LS)
    - Users add $r$ to private keys. LS adds $r'$ to corres. private keys
  - New $K_{LK} = K_{LK} + r + r'$
Handling Compromise of LS

Compromise $\forall i \ K'_U, H_U$ $\Rightarrow$ LS

X Confidentiality
X Authentication and Integrity
√ Spamming (limited)

Recovery: Re-instate LS. LM forces epoch change and update of MAC keys.

Simultaneous Compromise $K_{Uj};$ computes $K_L, K_U$ $\forall i$ $\Rightarrow$ $U_j$

√ Confidentiality (current epoch)
- enough to compromise $U_j$
X Authentication and Integrity
√ Spamming (limited)

Recovery: Requires LM to send e-mail to each user with updates for Private and MAC keys. Less expensive than re-establishing list

• If LM, LS compromised: re-establish list
• Insider threat: LS and User collusion
  • Difficult to detect
Implementation

- **SELS Prototype in Java**
  - Integrated with Eudora E-mail Client via command-line interface
  - Integrated with GnuPG Toolkit for standard Signature and Encryption operations

- **Components**
  - Crypto Utilities – SELS Enc/Dec, base64 encoding, random number generation, MACing, interface to GnuPG
  - List Mgmt – creating lists, subscribe users, manage lists
  - E-mail processing – process plaintext e-mail
  - Eudora Interface – send/receive e-mails

- **Work in Progress**
  - Plugin for Eudora, JavaMail
  - Integration with list server software; e.g., Mailman
Conclusions and Future Work

- Increasing use of ELS for exchanging private content ⇒ security is important

- Security for ELSs is significantly different from that for TPEE

- Provide solutions for confidentiality, integrity, authentication, and anti-spamming
  - E-mail plaintext not exposed at LS via proxy encryption

- Prototype implementation underway
  - Will develop plugins for Eudora, JavaMail
  - Will integrate SELS with common list server software; e.g., Mailman
Questions?