Internet Security

Enhanced Security Services for S/MIME

Thomas Göttlicher

April 20, 2004

Agenda

- Basics
- Technical
- Signed receipts
- Security labels
- Secure mailing lists
- Signed certificates

1 Basics

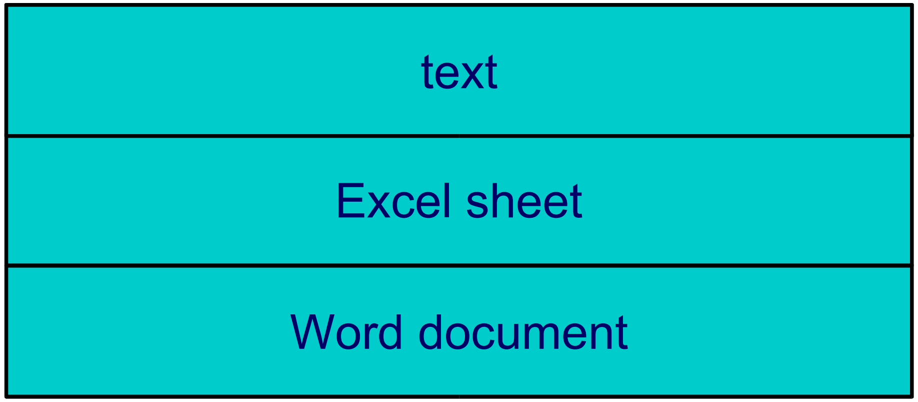
Basics

- S/MIME = Secure MIME
- protect MIME e-mail

Basics

- S/MIME = Secure MIME
- protect MIME e-mail

MIME e-mail



text

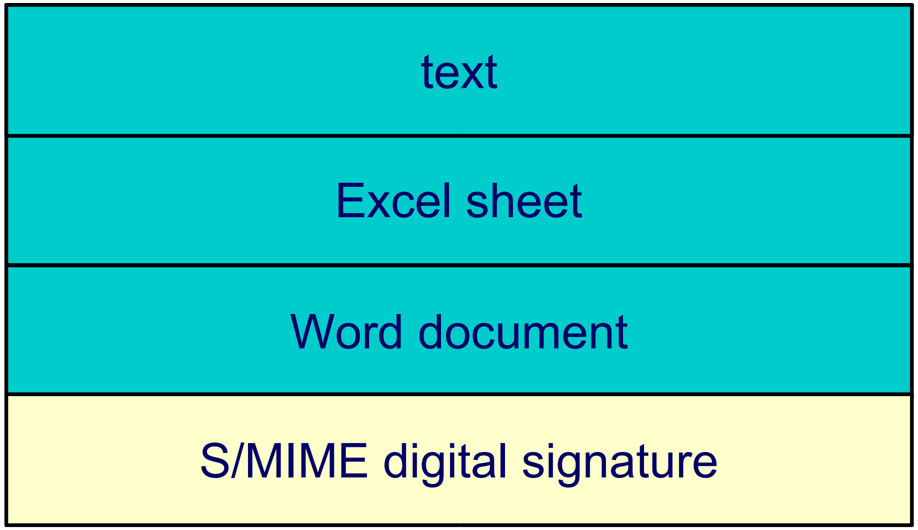
Excel sheet

Word document

Basics

- S/MIME = Secure MIME
- protect MIME e-mail

signed S/MIME e-mail



text

Excel sheet

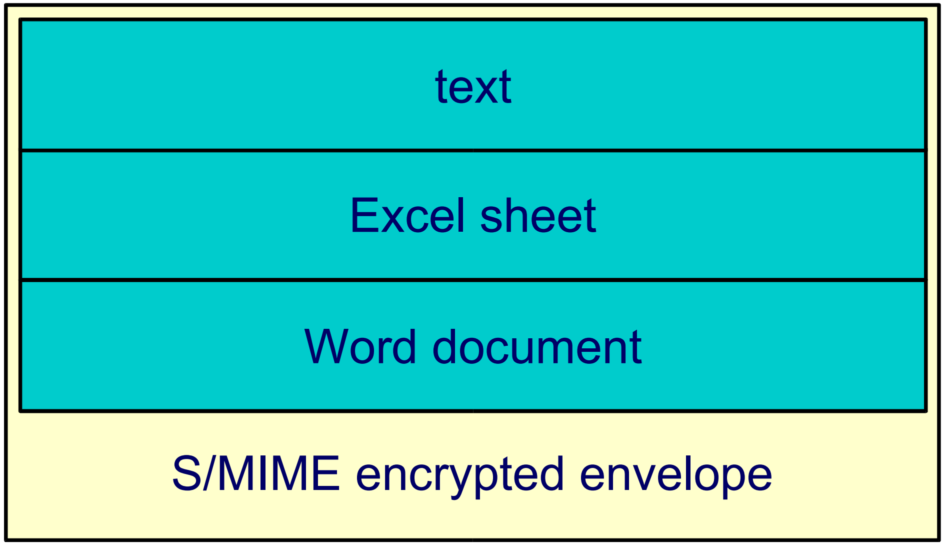
Word document

S/MIME digital signature

Basics

- S/MIME = Secure MIME
- protect MIME e-mail

encrypted S/MIME e-mail



text

Excel sheet

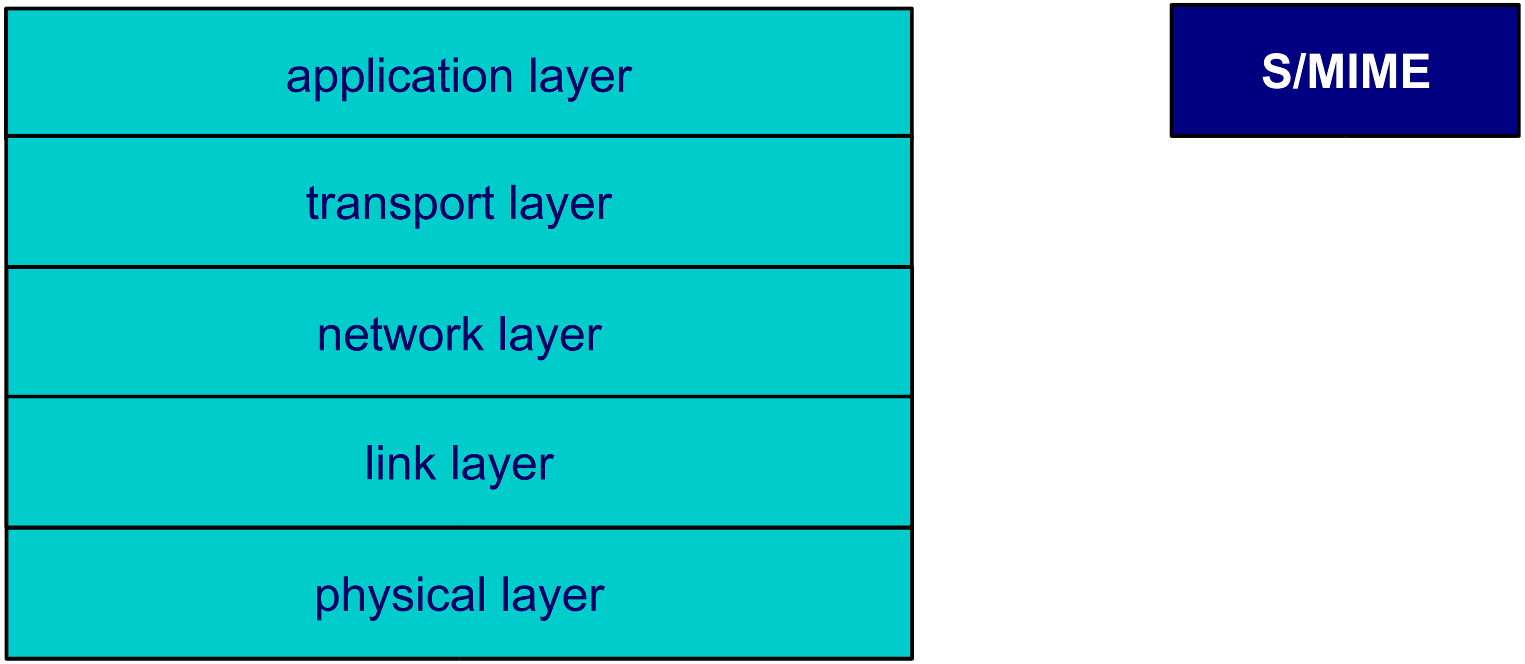
Word document

S/MIME encrypted envelope

2 Technical

- Internet Layer
- Compatibility
- Triple Wrapping

Internet Layer



application layer

transport layer

network layer

link layer

physical layer

S/MIME

Compatibility

- S/MIME v3 can read messages from S/MIME v2

- BUT: S/MIME v3 messages are unreadable by S/MIME v2

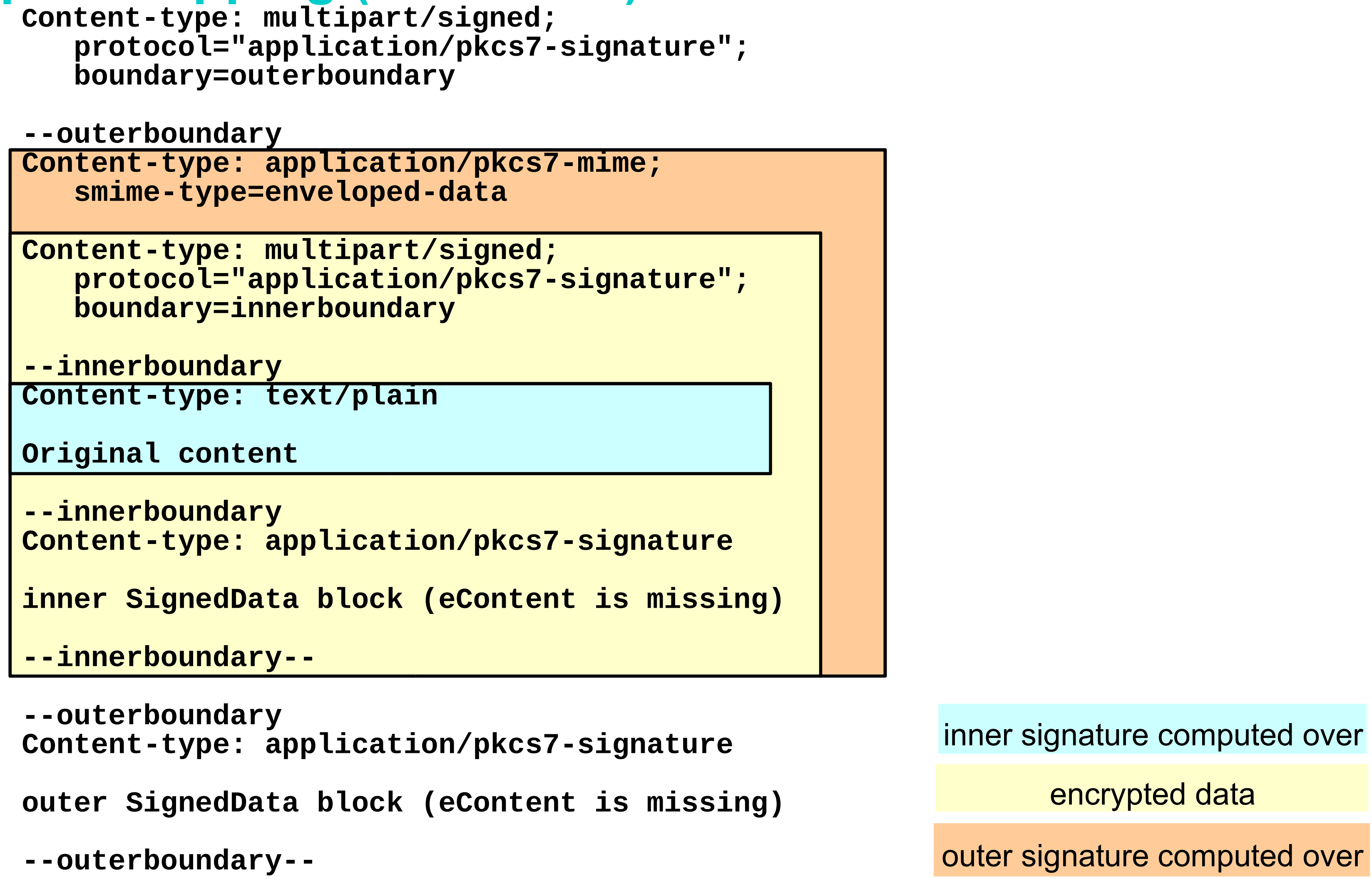
Triple Wrapping

- Message has been signed, encrypted and signed again
- Inside signature: content integrity
- Encrypted body: confidentiality
- Outside signature: integrity for information produced hop-by-hop

Triple Wrapping (continued)

```
Content-type: multipart/signed;
   protocol="application/pkcs7-signature";
   boundary=outerboundary

--outerboundary
Content-type: application/pkcs7-mime;
   smime-type=enveloped-data

Content-type: multipart/signed;
   protocol="application/pkcs7-signature";
   boundary=innerboundary

--innerboundary
Content-type: text/plain

Original content

--innerboundary
Content-type: application/pkcs7-signature

inner SignedData block (eContent is missing)

--innerboundary--

--outerboundary
Content-type: application/pkcs7-signature

outer SignedData block (eContent is missing)

--outerboundary--
```

Triple Wrapping (continued)



```
Content-type: multipart/signed;
   protocol="application/pkcs7-signature";
   boundary=outerboundary

--outerboundary
Content-type: application/pkcs7-mime;
   smime-type=enveloped-data

Content-type: multipart/signed;
   protocol="application/pkcs7-signature";
   boundary=innerboundary

--innerboundary
Content-type: text/plain

Original content

--innerboundary
Content-type: application/pkcs7-signature

inner SignedData block (eContent is missing)

--innerboundary--

--outerboundary
Content-type: application/pkcs7-signature

outer SignedData block (eContent is missing)

--outerboundary--
```

inner signature computed over

encrypted data

outer signature computed over

3 Signed Receipts

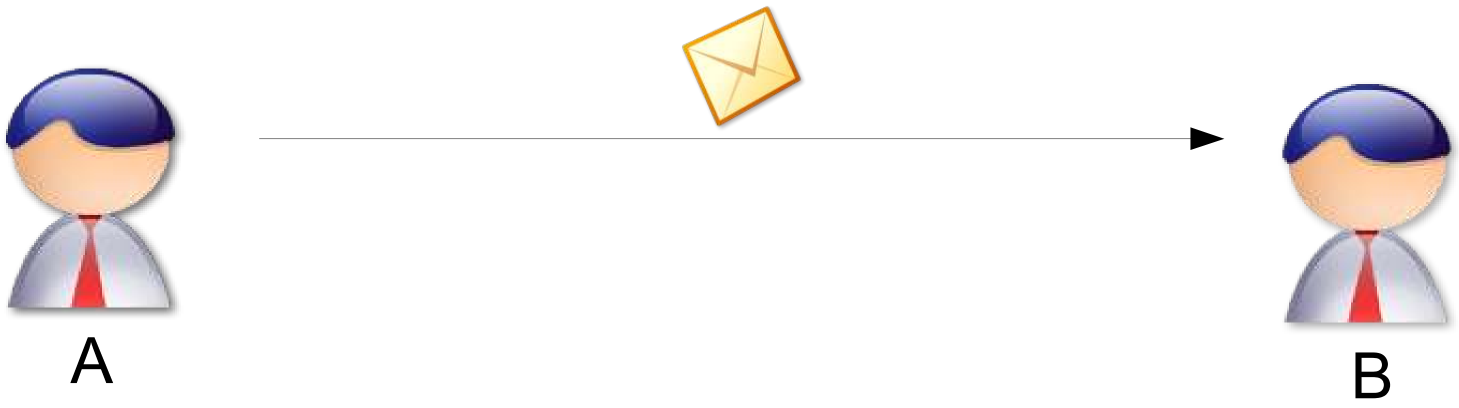
Signed Receipts

- Proof of delivery of a message

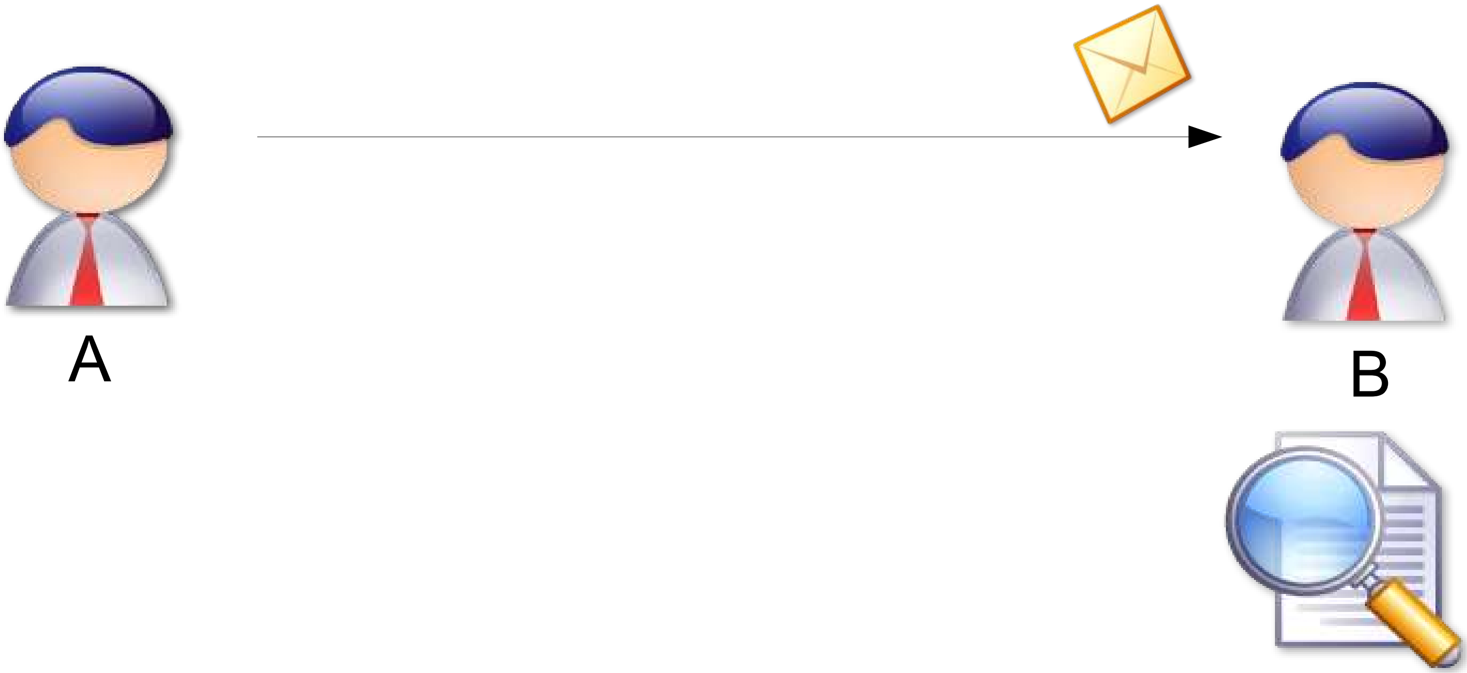
- Before processing a receipt-request: the receiving agent must verify the signature
  => no receipt if signature is invalid

- Receiving user agent software should automatically create a signed receipt when requested

Signed Receipts (Example)



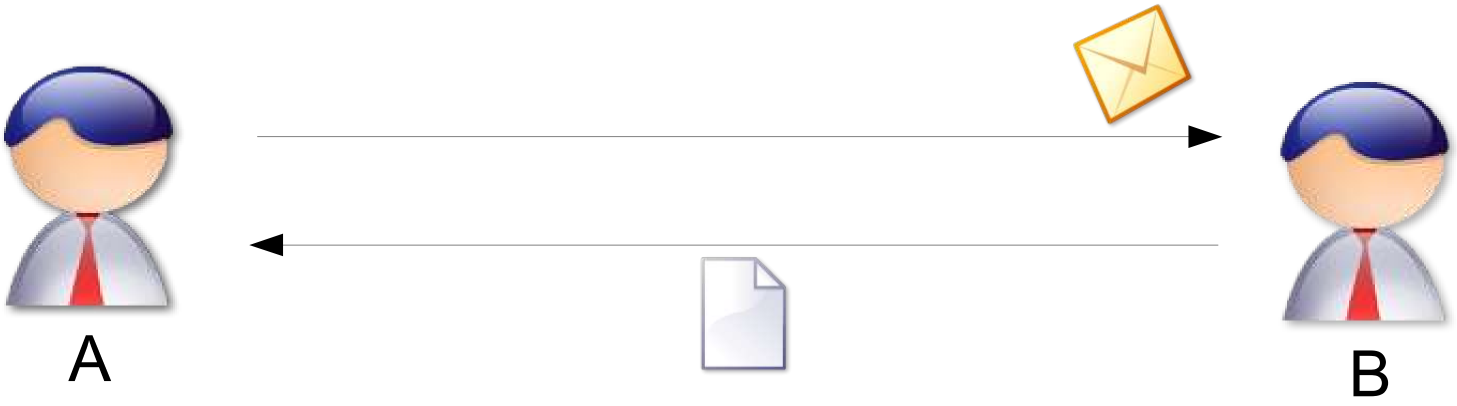
Signed Receipts (Example)



A

B

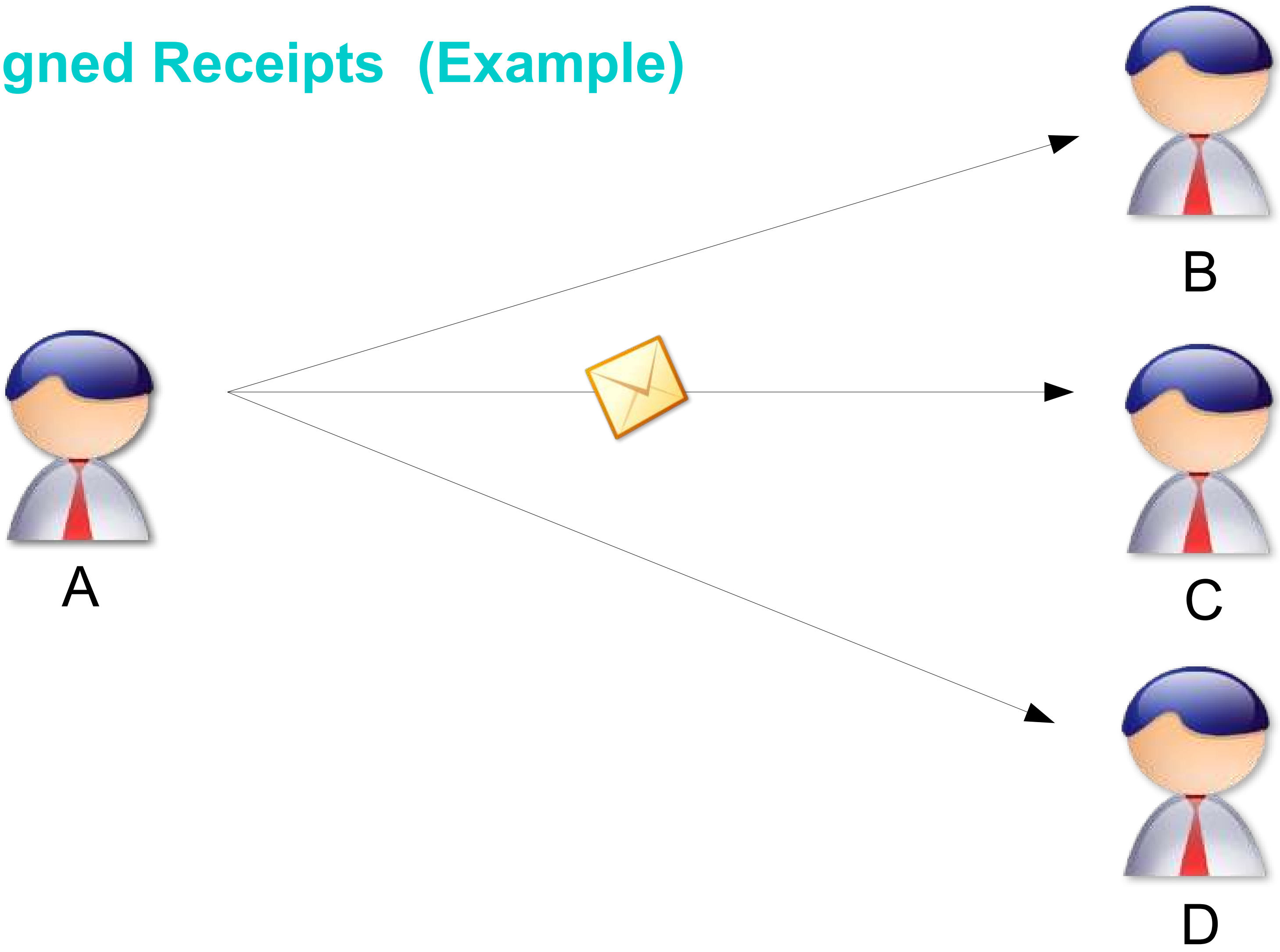
Signed Receipts (Example)



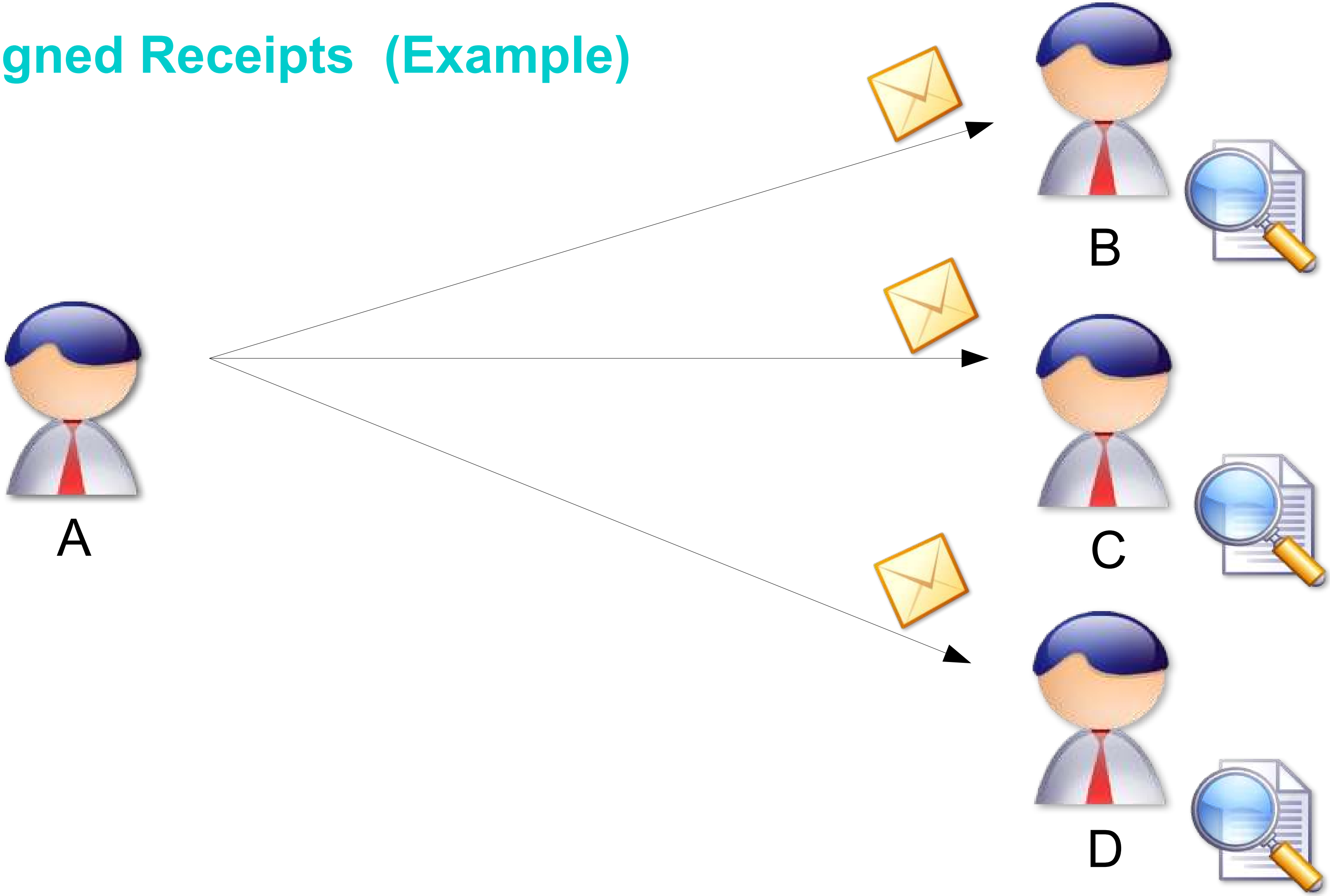
Signed Receipts (continued)

- Receipts can be requested from
  - all recipients

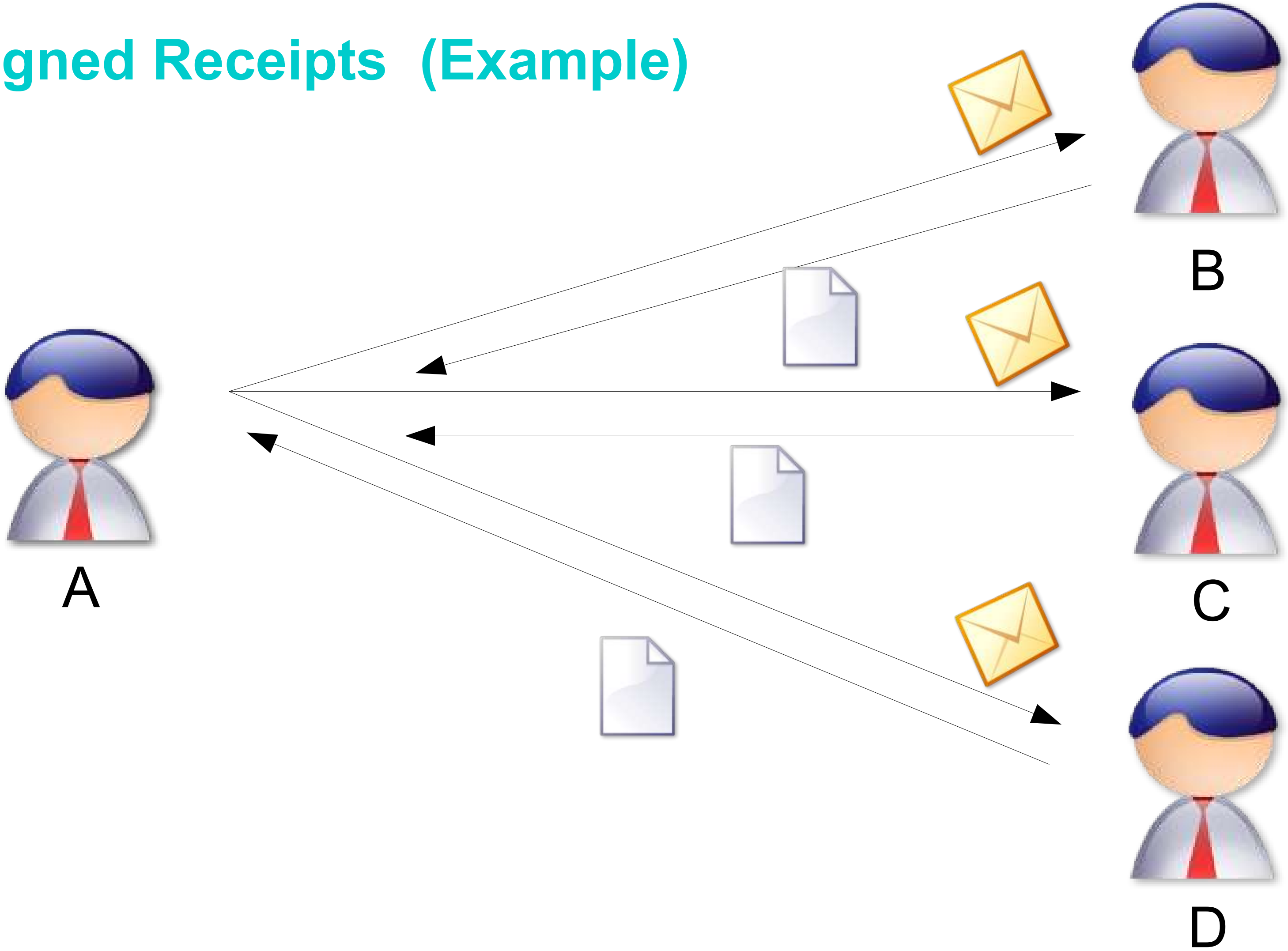
Signed Receipts (Example)



Signed Receipts (Example)



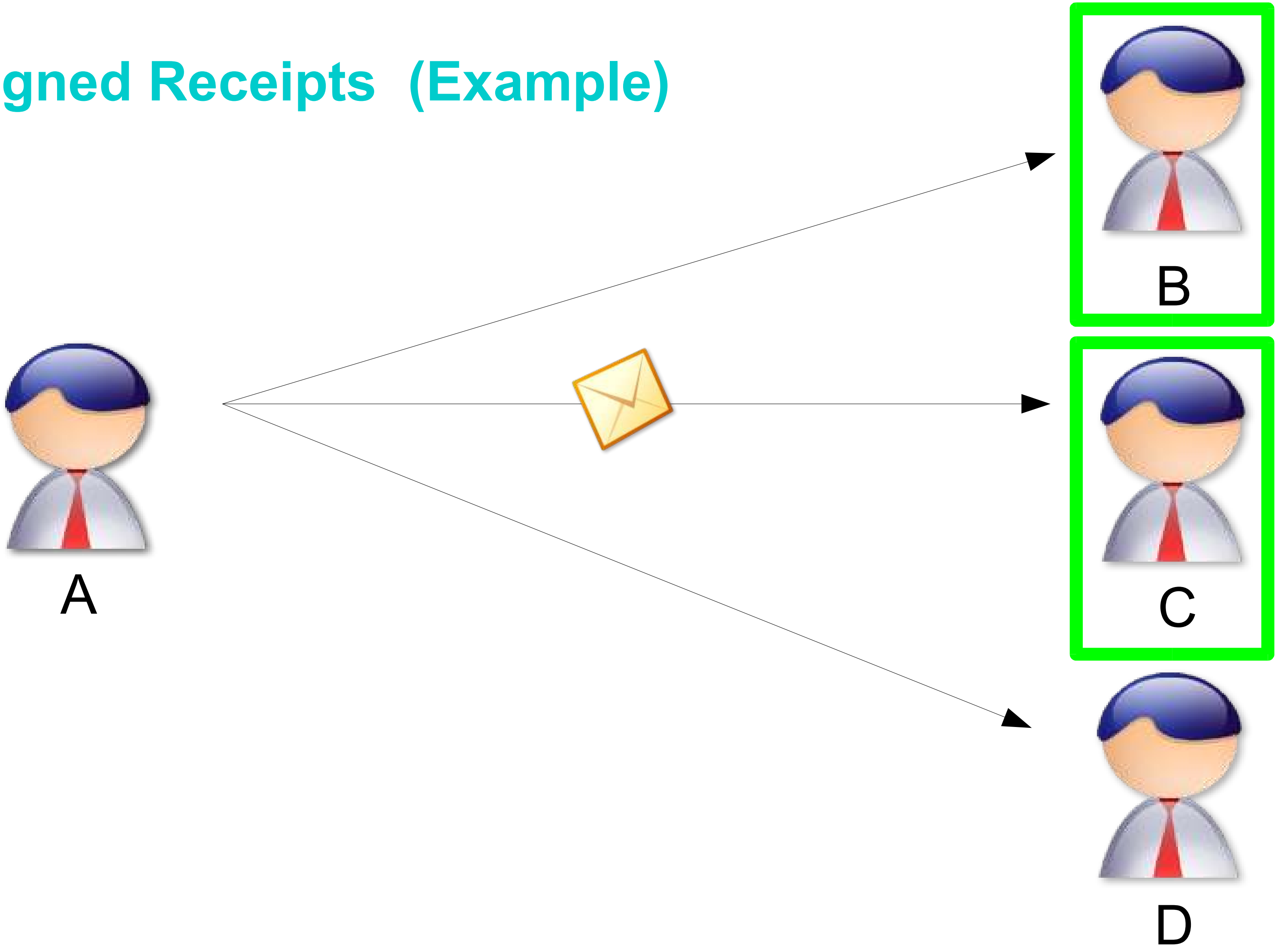
Signed Receipts (Example)



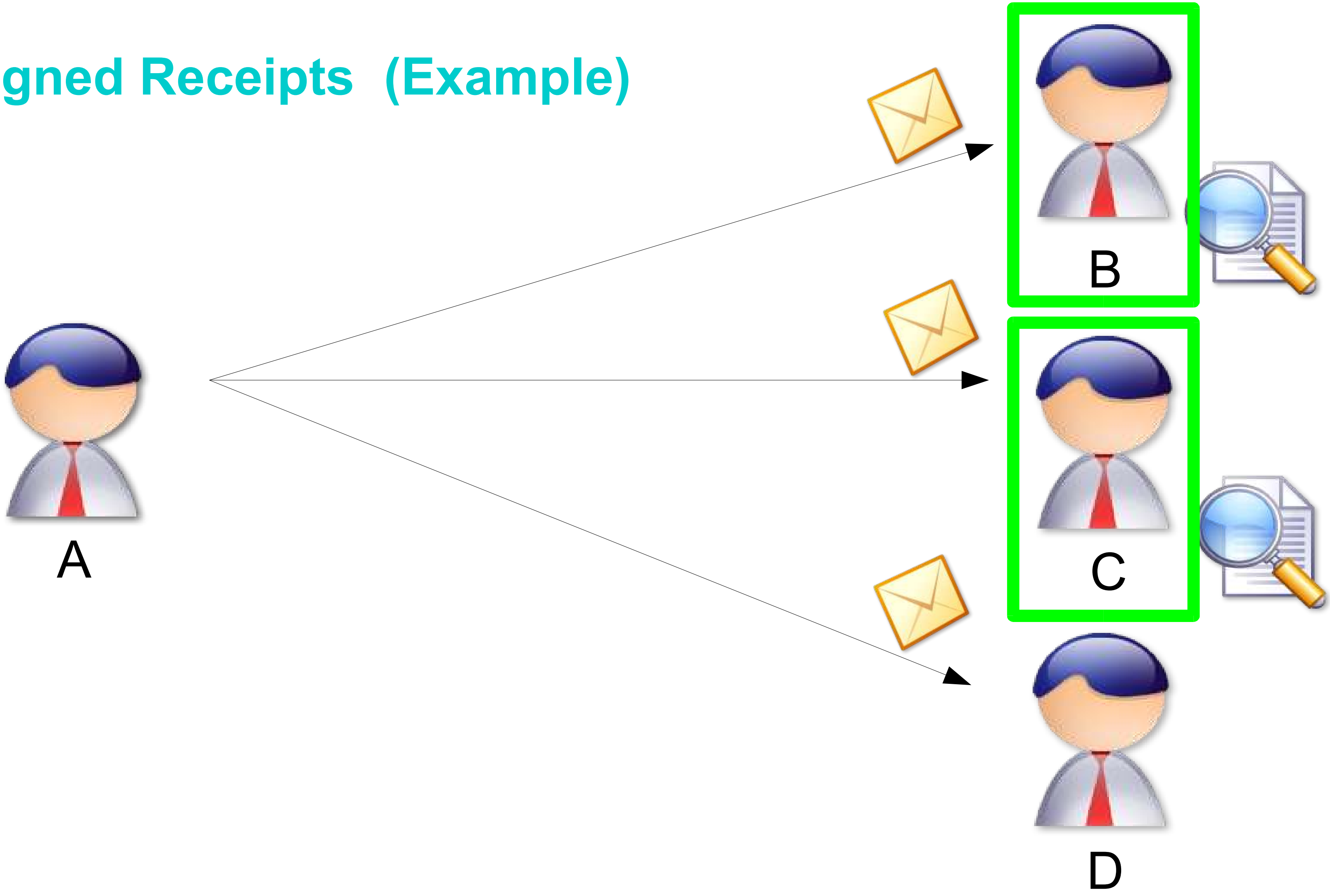
Signed Receipts (continued)

- Receipts can be requested from
  - all recipients
  - a specific list of recipients

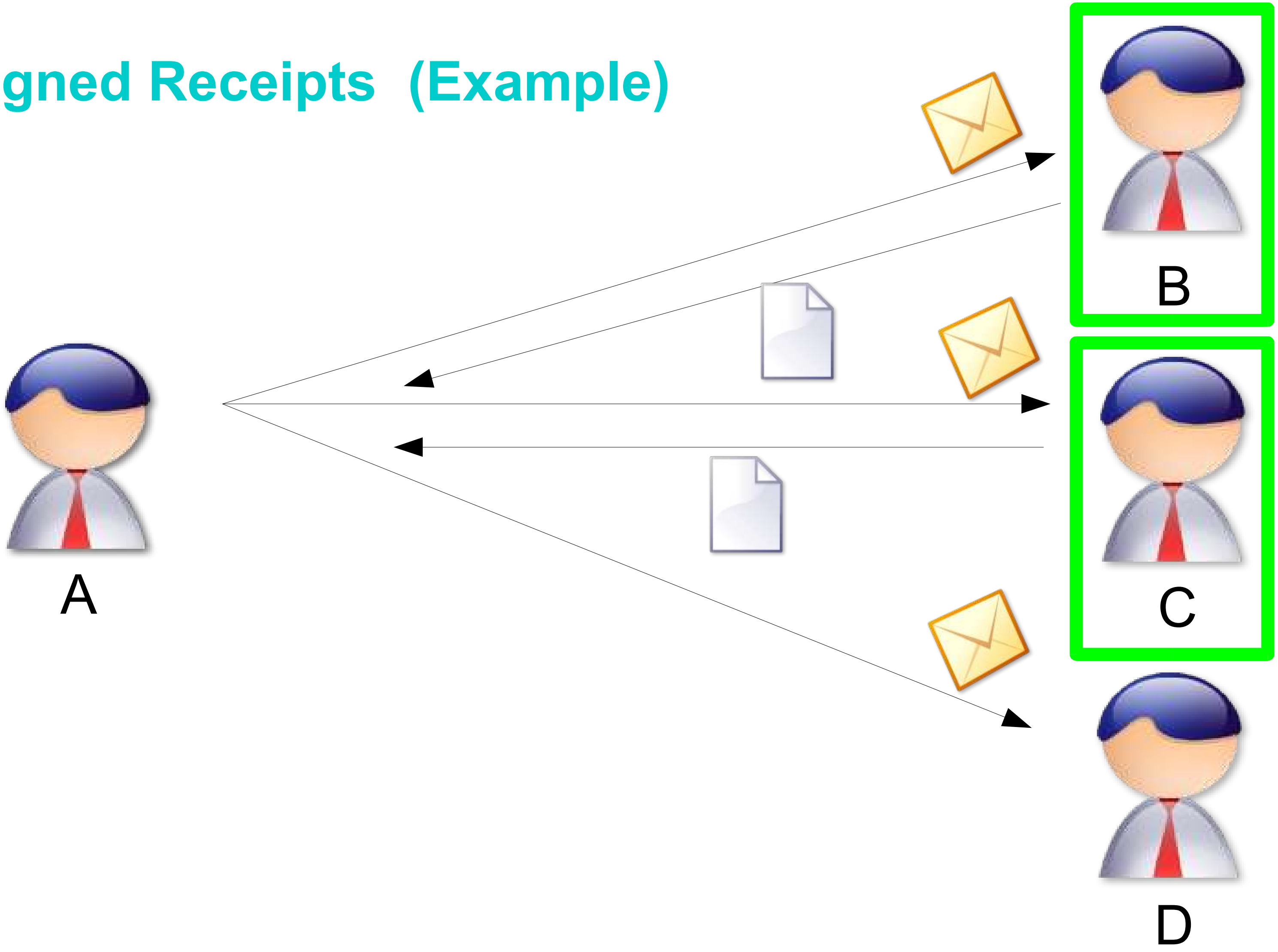
Signed Receipts (Example)



Signed Receipts (Example)



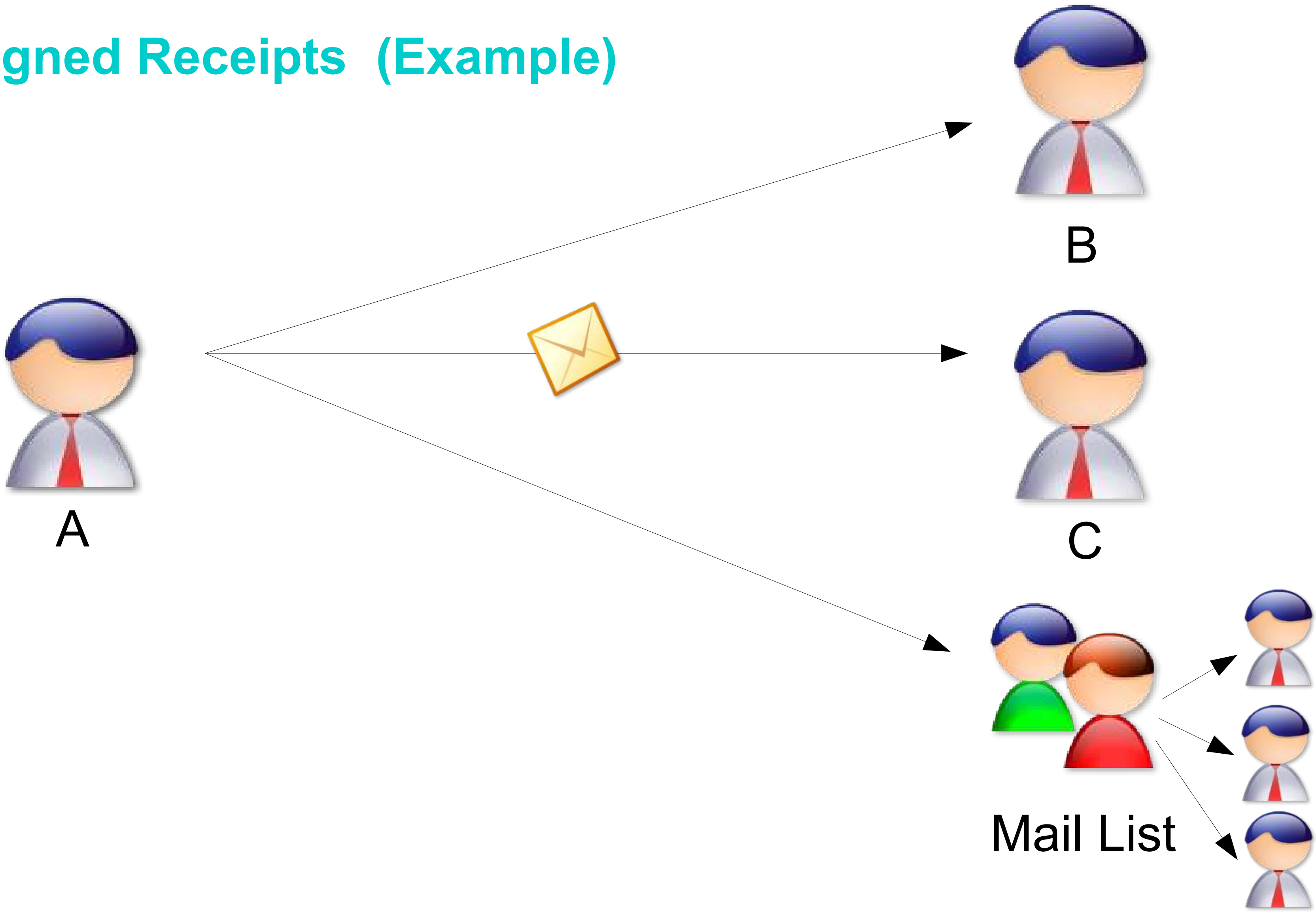
Signed Receipts (Example)



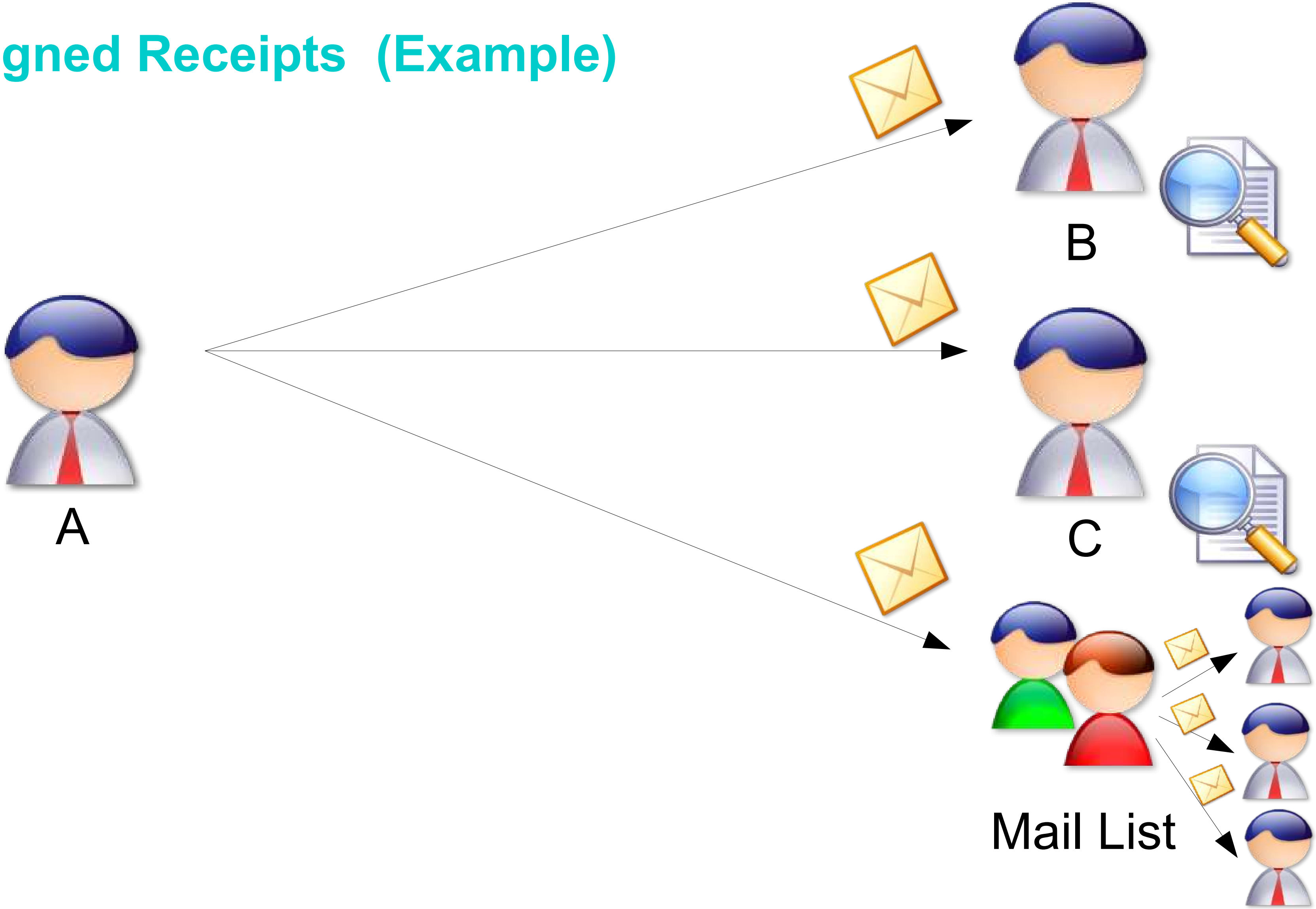
Signed Receipts (continued)

- Receipts can be requested from
  - all recipients
  - a specific list of recipients
  - first tier (= recipients that did not receive the message as members of a mailing list)

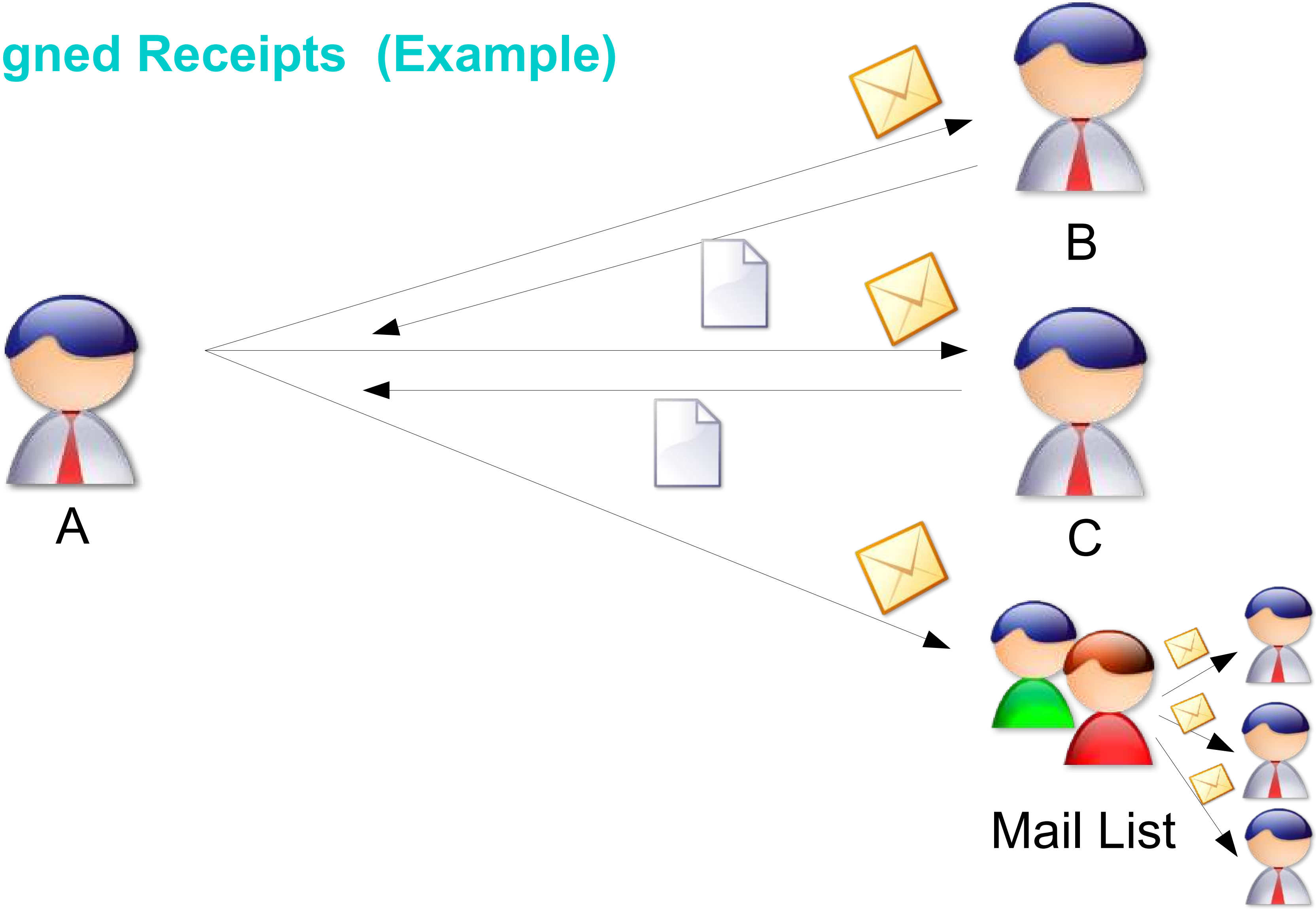
Signed Receipts (Example)



Signed Receipts (Example)



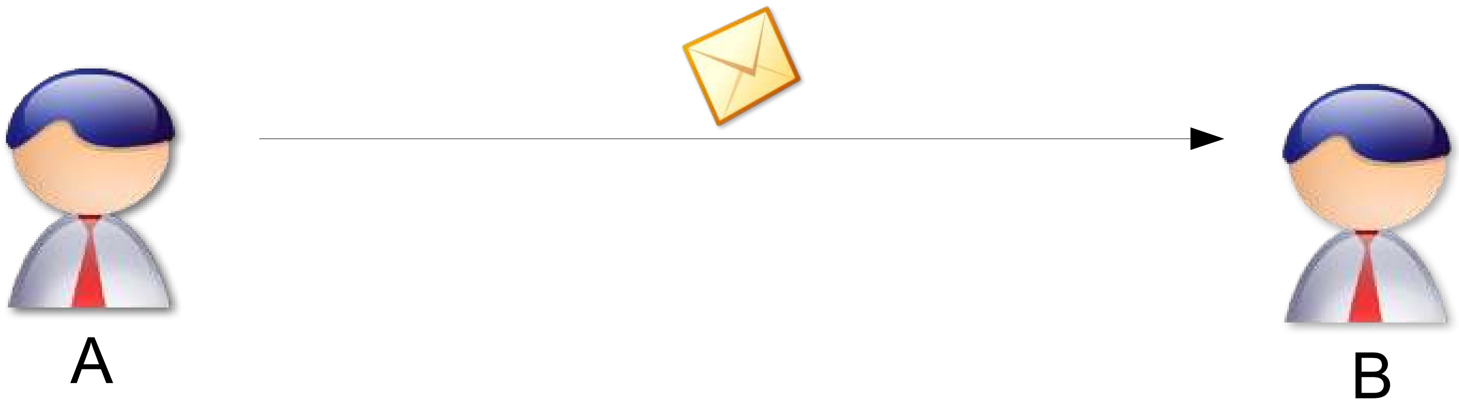
Signed Receipts (Example)



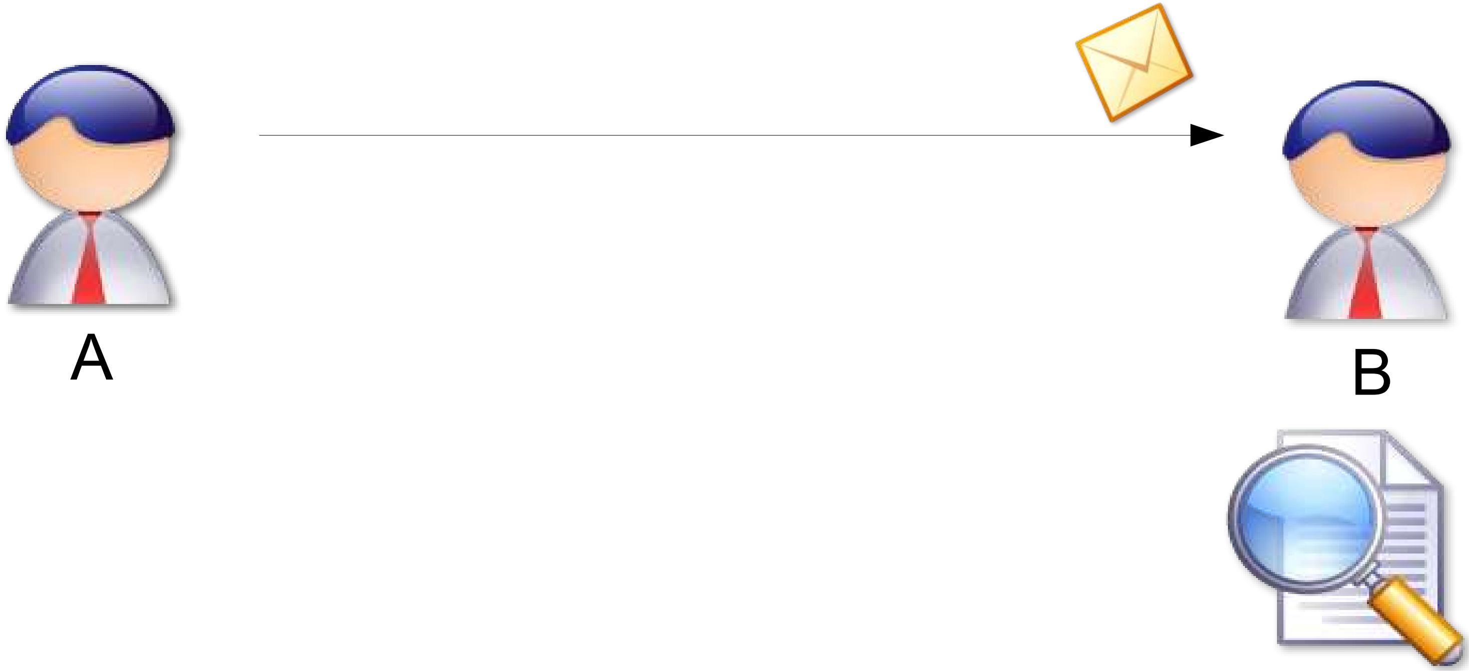
Signed Receipts (continued)

- Receipts can be requested from
  - all recipients
  - a specific list of recipients
  - first tier (= recipients that did not receive the message as members of a mailing list)
- Sender can indicate that receipts be sent to many places
  - receipt not just to the sender

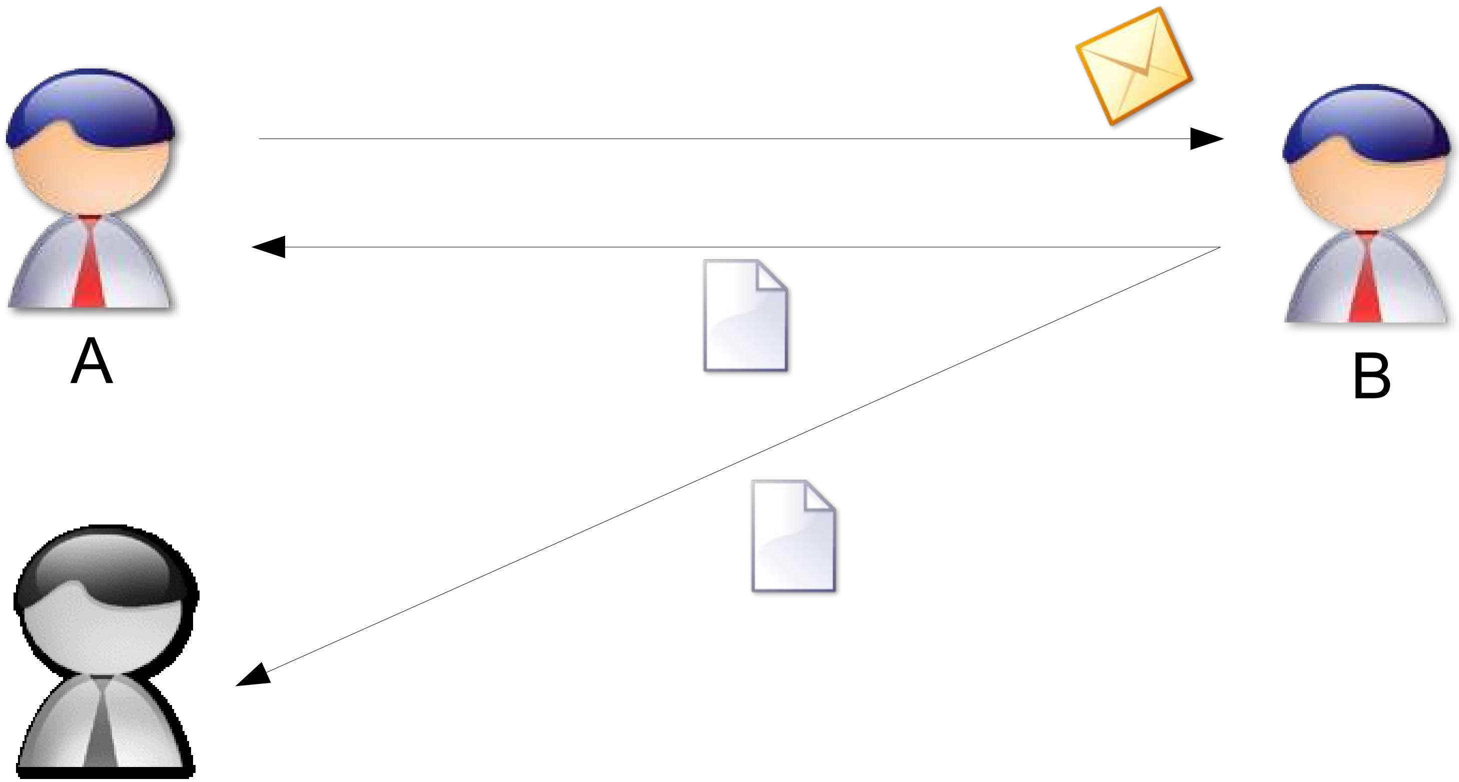
Signed Receipts (Example)



Signed Receipts (Example)



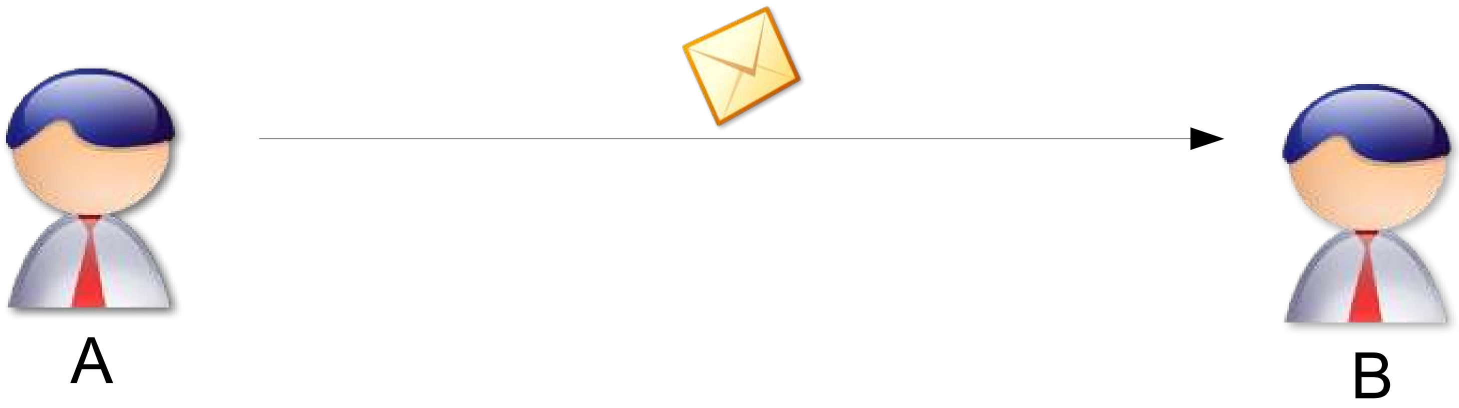
Signed Receipts (Example)



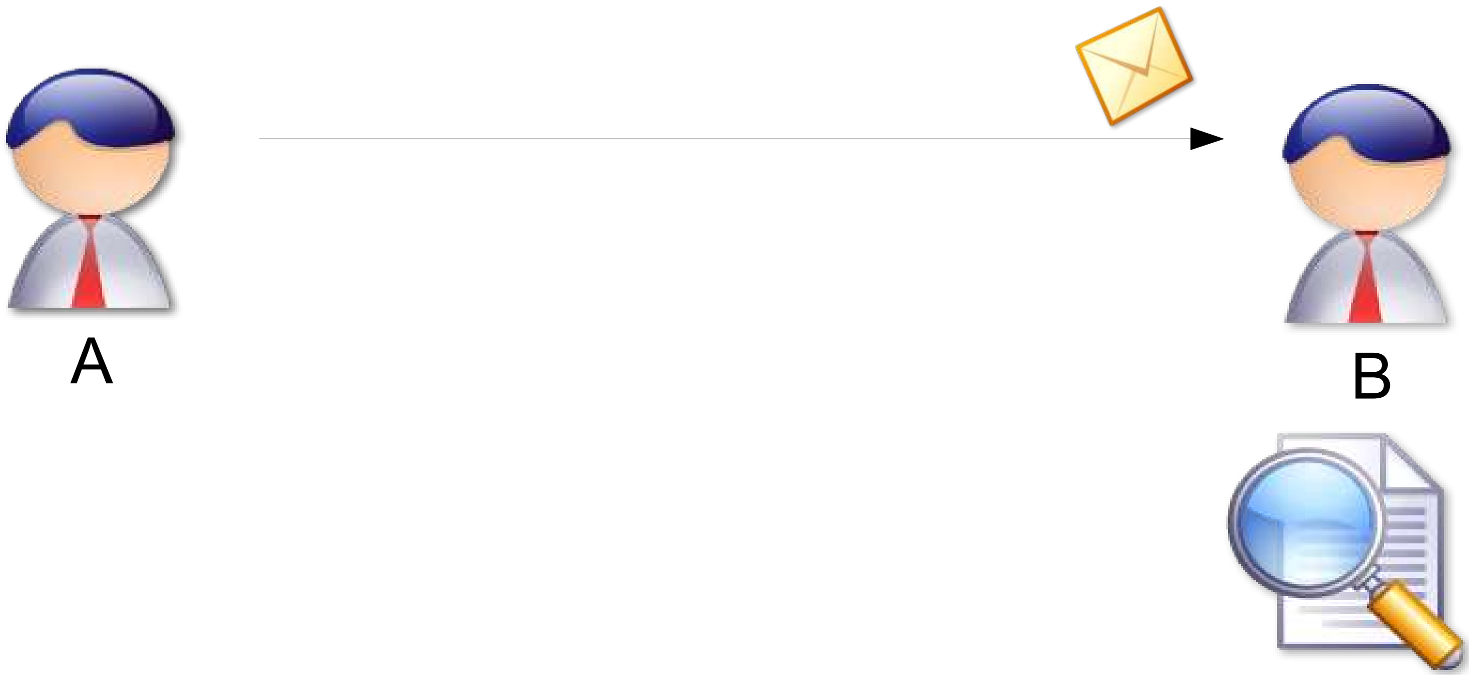
Signed Receipts (continued)

- Receipts can be requested from
  - all recipients
  - a specific list of recipients
  - first tier (= recipients that did not receive the message as members of a mailing list)
- Sender can indicate that receipts be sent to many places
  - receipt not just to the sender
  - not even to the sender

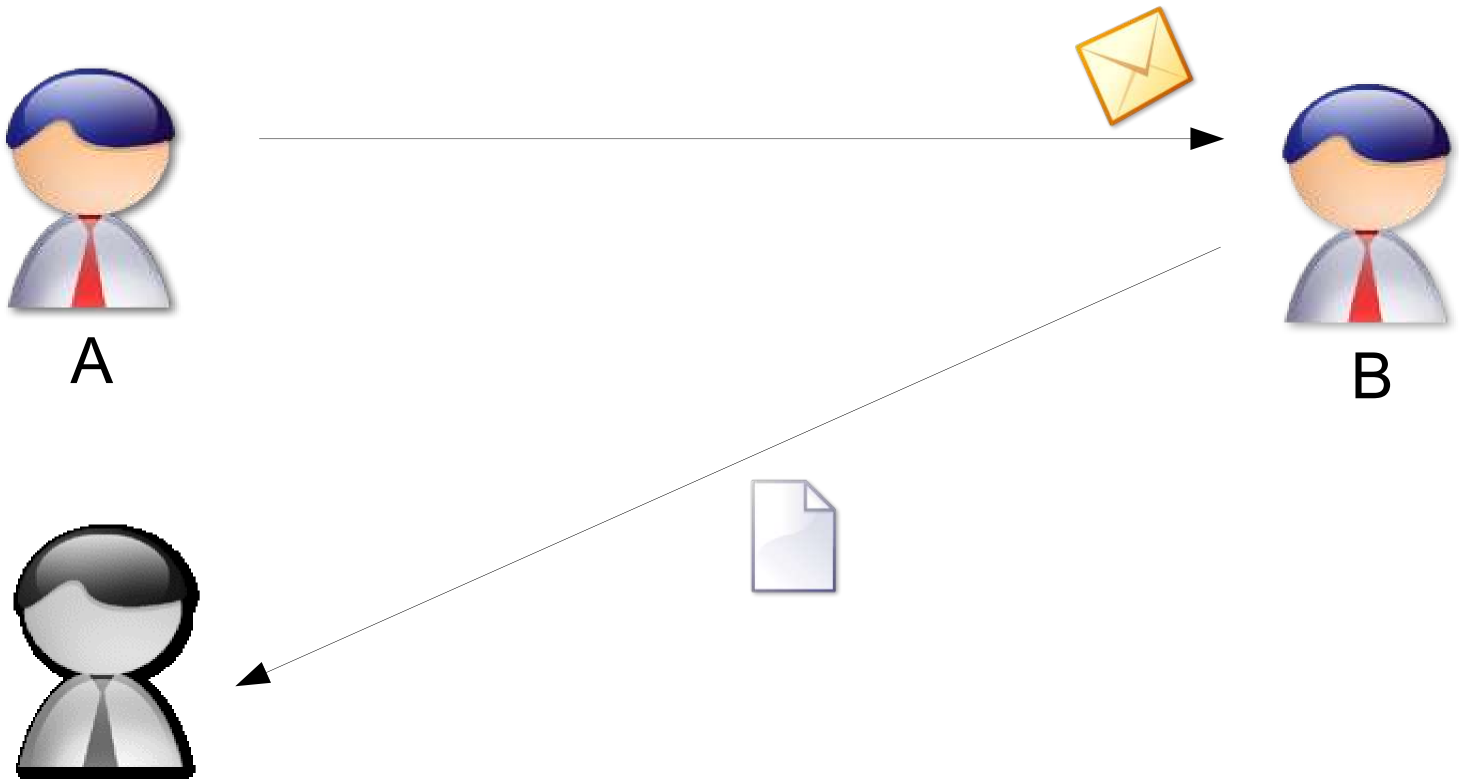
Signed Receipts (Example)



Signed Receipts (Example)



Signed Receipts (Example)



Signed Receipts (continued)

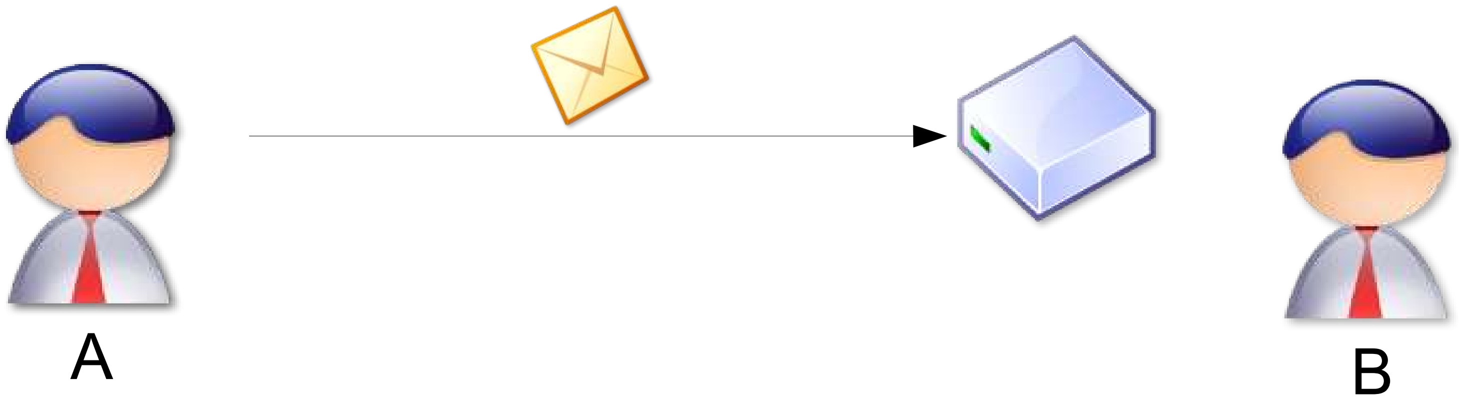
- Receipts can be requested from
  - all recipients
  - a specific list of recipients
  - first tier (= recipients that did not receive the message as members of a mailing list)
- Sender can indicate that receipts be sent to many places
  - receipt not just to the sender
  - not even to the sender
- Multiple Receipt Requests: Each recipient should only return one receipt
- No singed receipt for a signed receipt

4 Security Labels

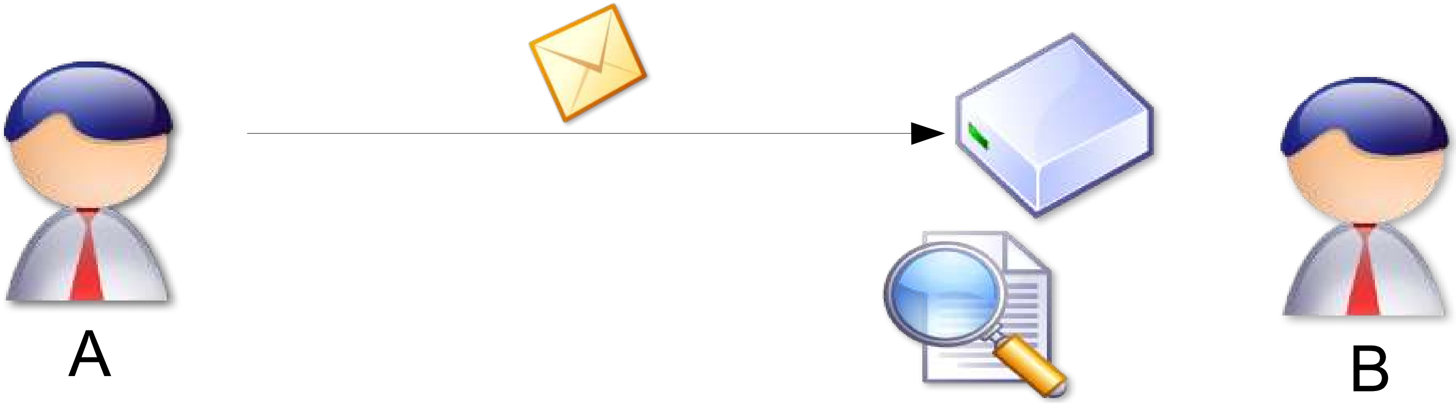
Security Labels

- Set of security information regarding the sensitivity of the content that is protected by S/MIME encapsulation
- Access control: receiving agent examines the security labels and determines whether or not the recipient is allowed to see the contents
- Security Labels must be signed attributes
- Signature must be verified and valid, before processing a security label
- Classification: unmarked, unclassified, restricted, confidential, secret, top-secret; other values can be defined by any organization

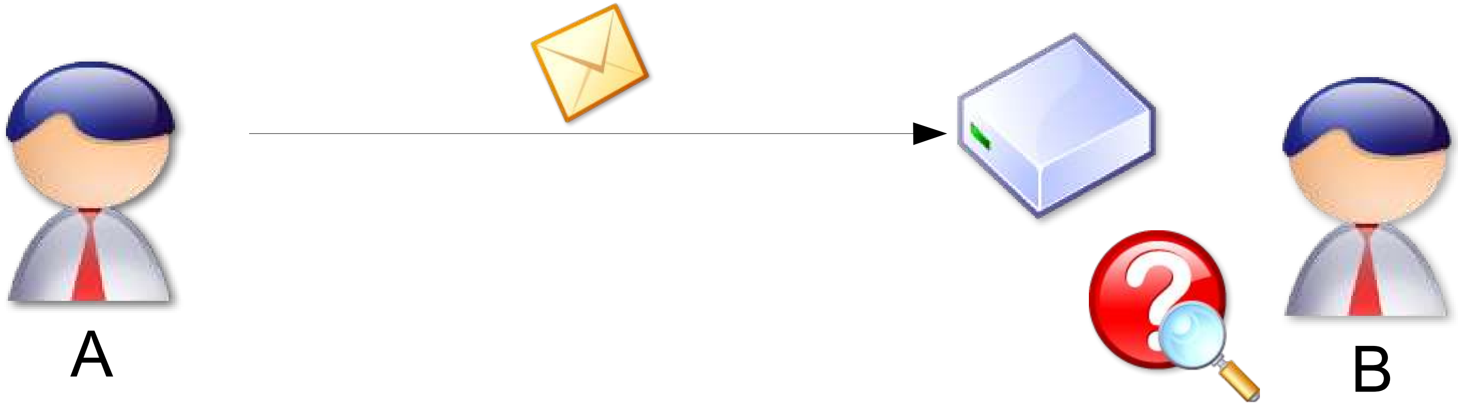
Security Labels (Example)



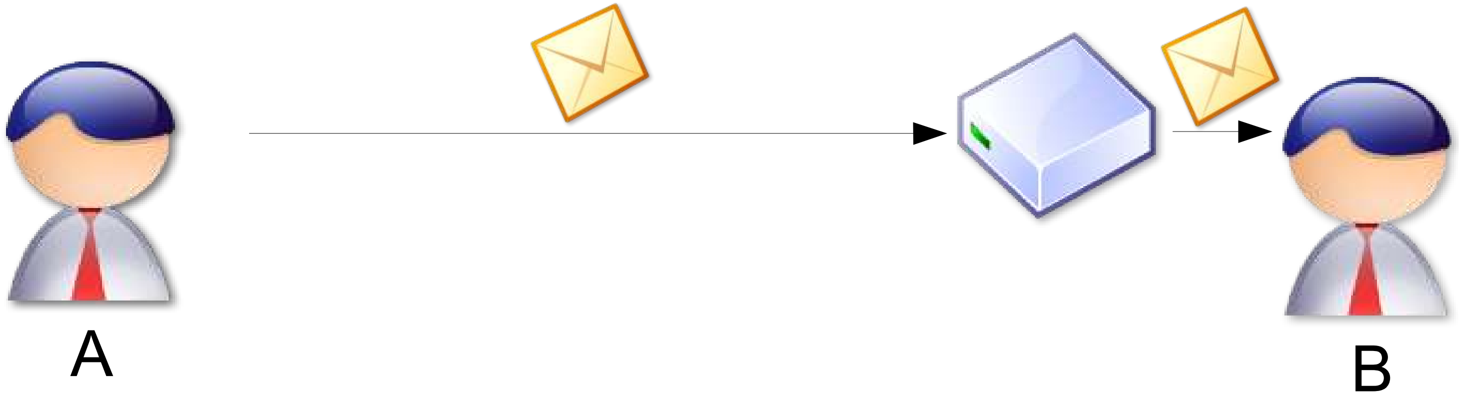
Security Labels (Example)



Security Labels (Example)



Security Labels (Example)



Equivalent Security Labels

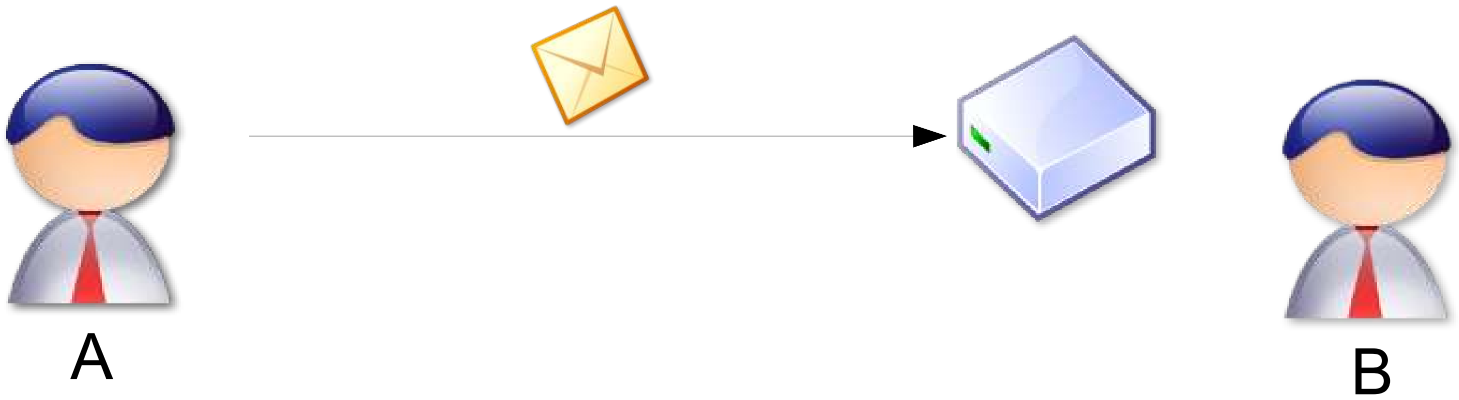
- Organizations are allowed to define their own security policies, many different security policies will exist
  => Equivalences between different security policies of different organizations

- Receiving agents have the option to process EquivalentLabels attributes

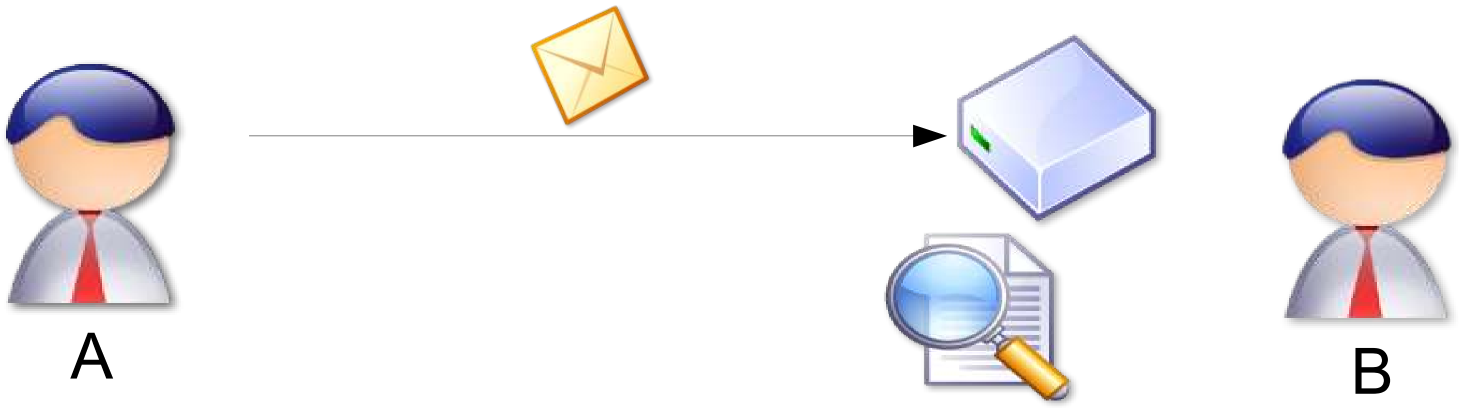
- Receiving agent processes equivalent labels only if it trusts the signer

- If the receiving agent understands the security label, it must ignore all equivalent labels

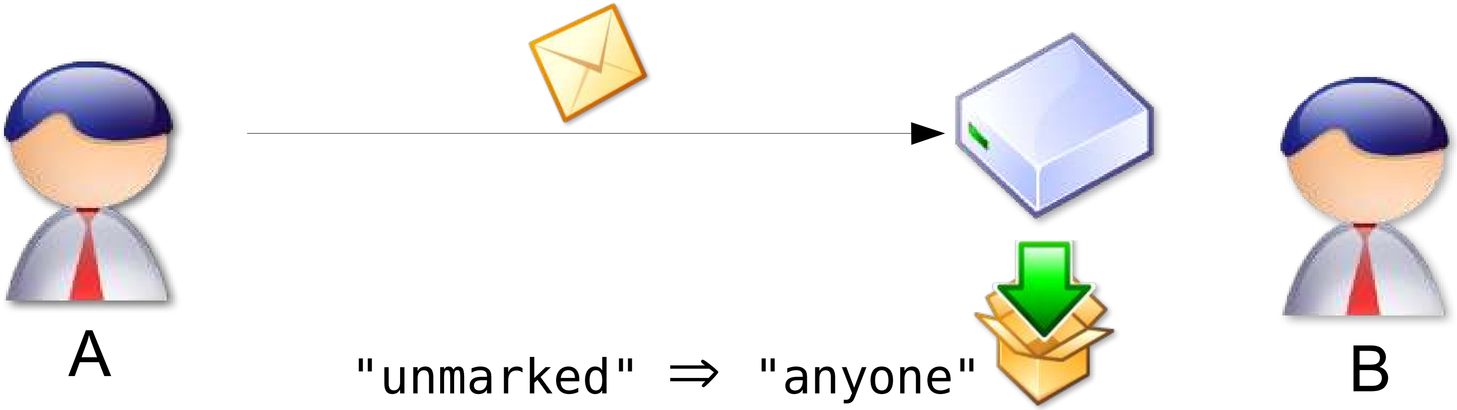
Security Labels (Example)



Security Labels (Example)



Security Labels (Example)

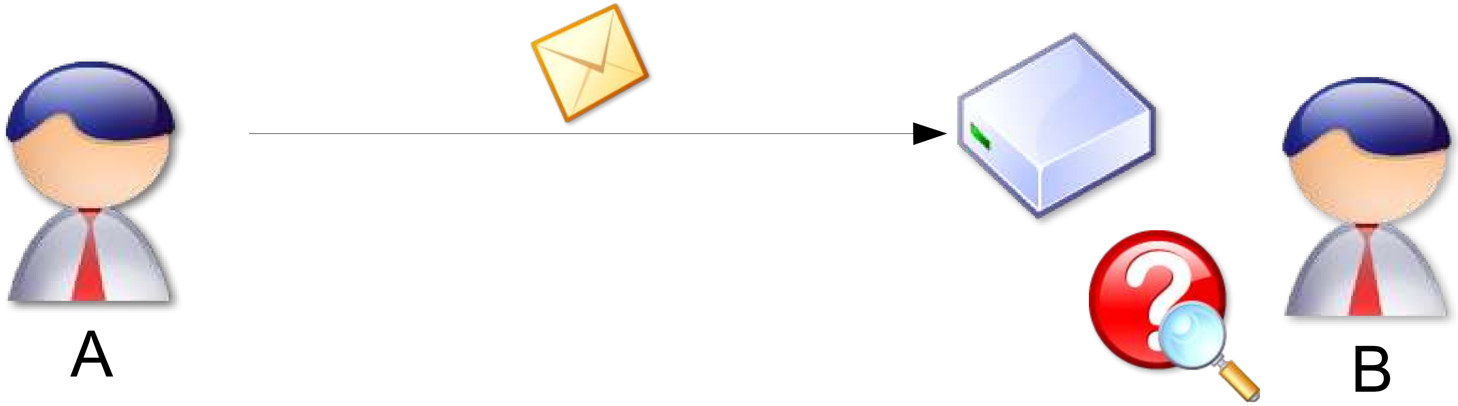


A

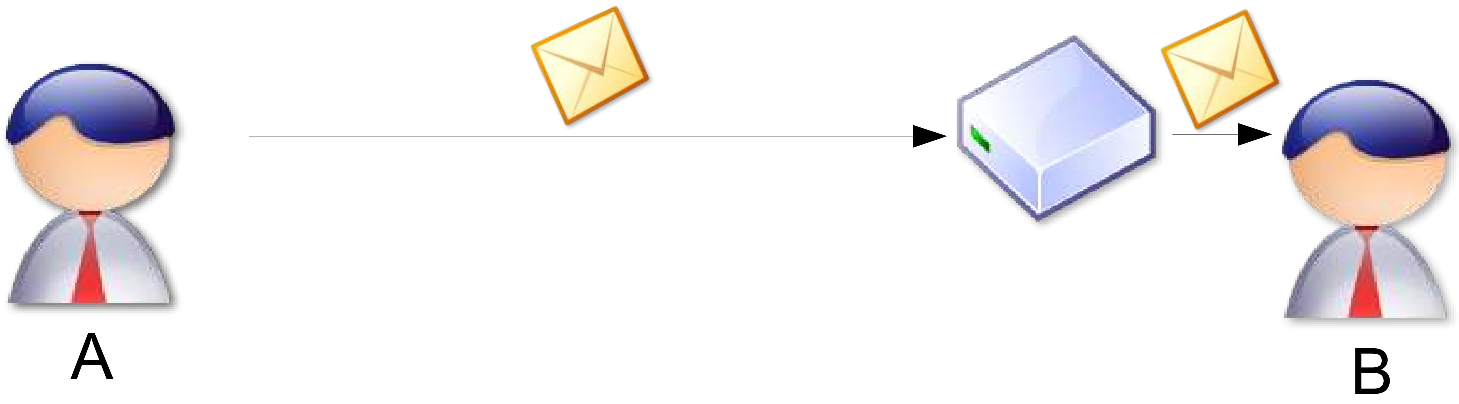
"unmarked" ⇒ "anyone"

B

Security Labels (Example)



Security Labels (Example)



5 Secure Mailing Lists

- Mail List Management
- Mail Loops
- Receipts

Mail List Management

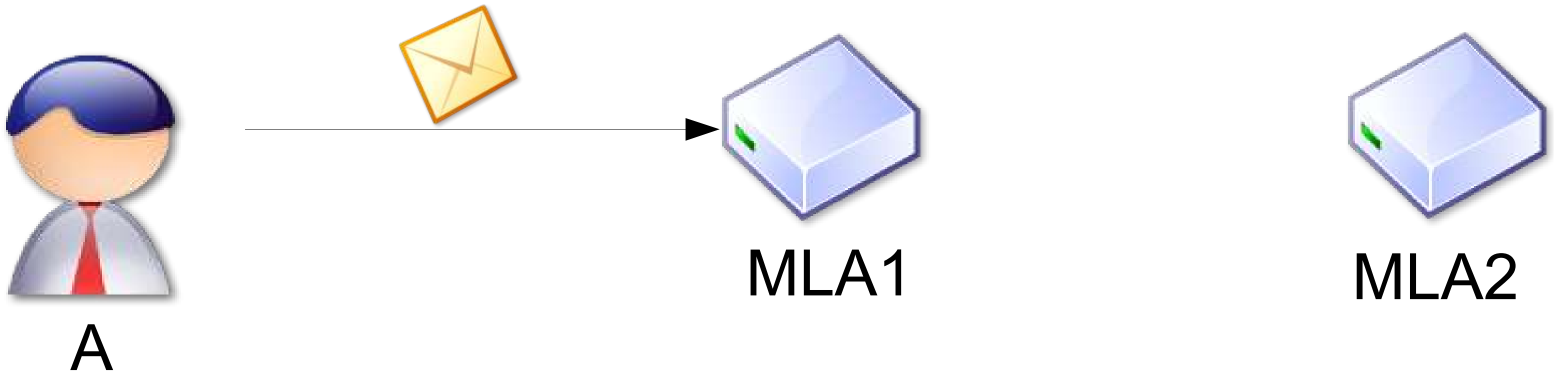
- Sending agents must create recipient-specific data structures for each recipient of an encrypted message.
- Large number of recipients => resources needs
- Mail List Agents (MLA) can take a singe message and perform the recipient-specific encryption

Mail List Management - Mail Loops

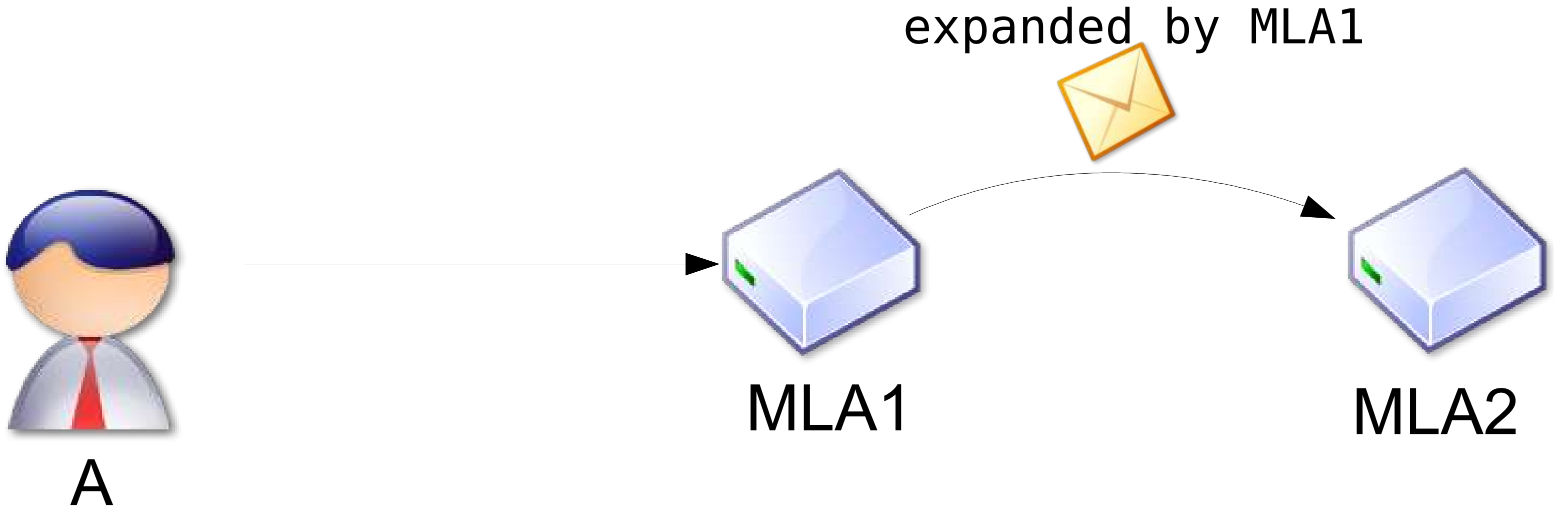
- One mailing list is member of a second and the second is member of the first.

- MLA have to prevent Mail loops
  - Each Time a MLA expands a message it adds its own identifier to the history
  - If own unique identifier is in the history
    => Mail loop
    - Don't send the message to the list again
    - Warning to a human mail list administrator

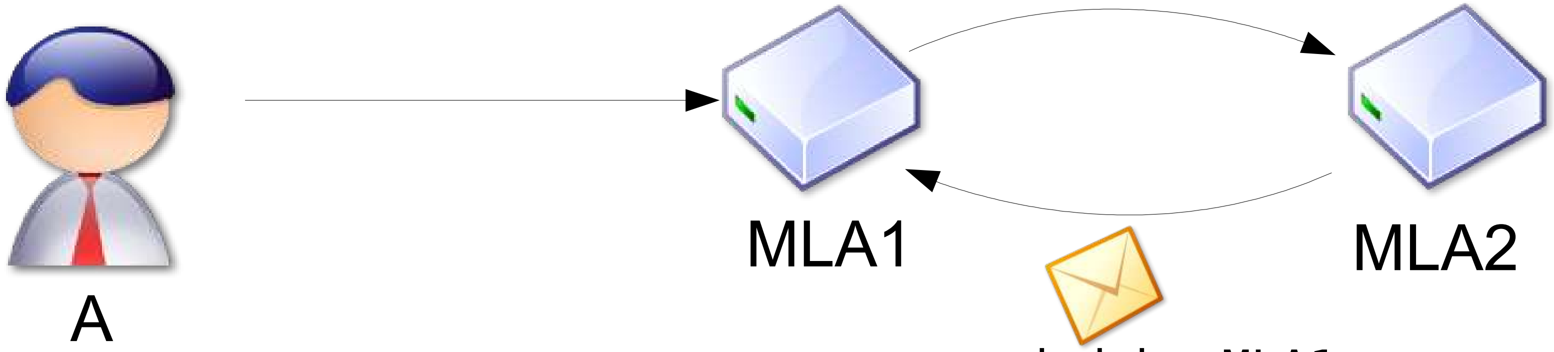
Mail List Management - Mail Loops (Example)



Mail List Management - Mail Loops (Example)

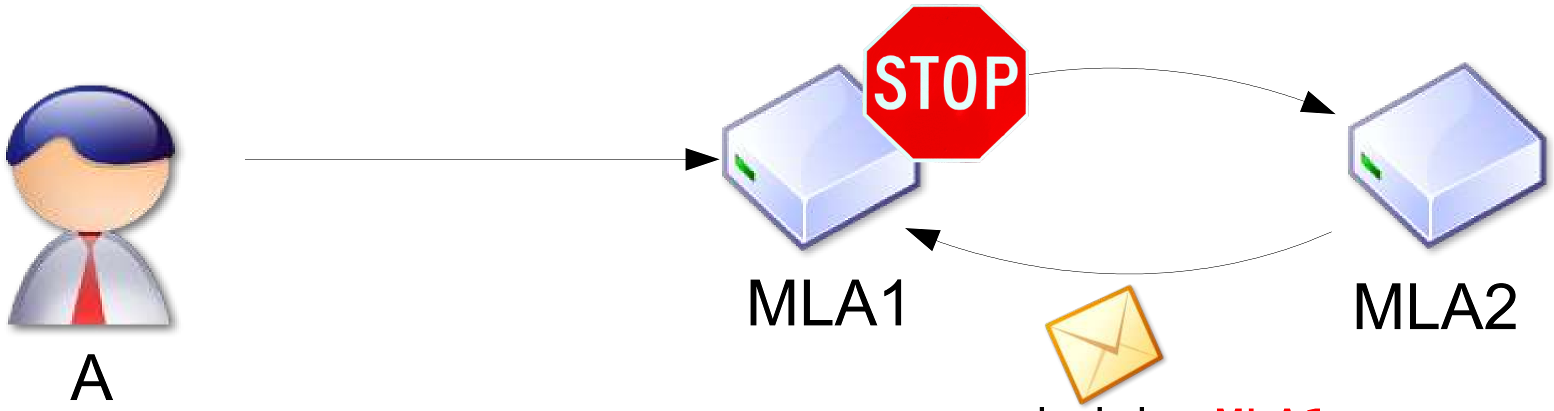


Mail List Management - Mail Loops (Example)



```
expanded by MLA1
expanded by MLA2
```

Mail List Management - Mail Loops (Example)



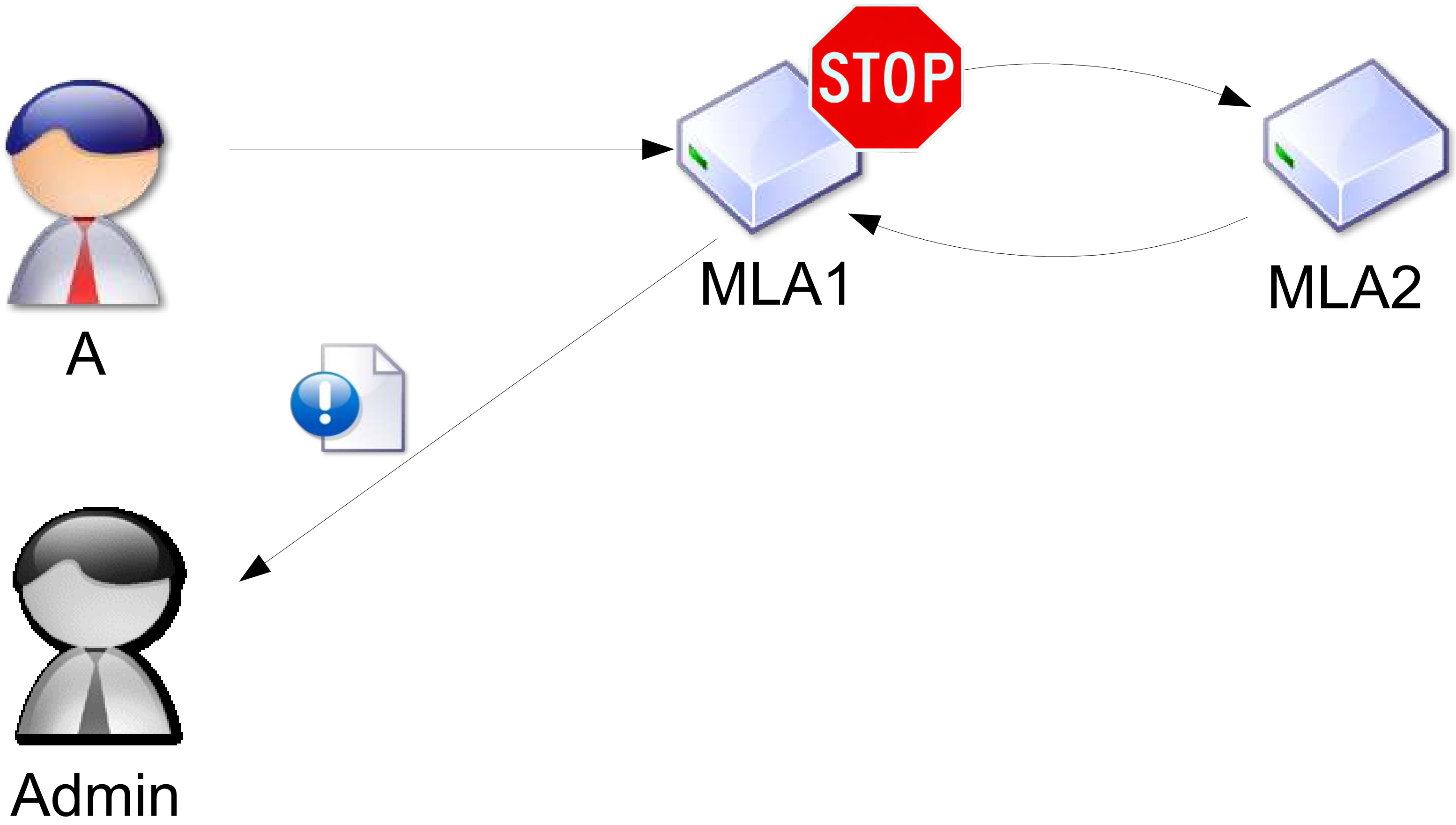
A

MLA1

MLA2

expanded by MLA1
expanded by MLA2

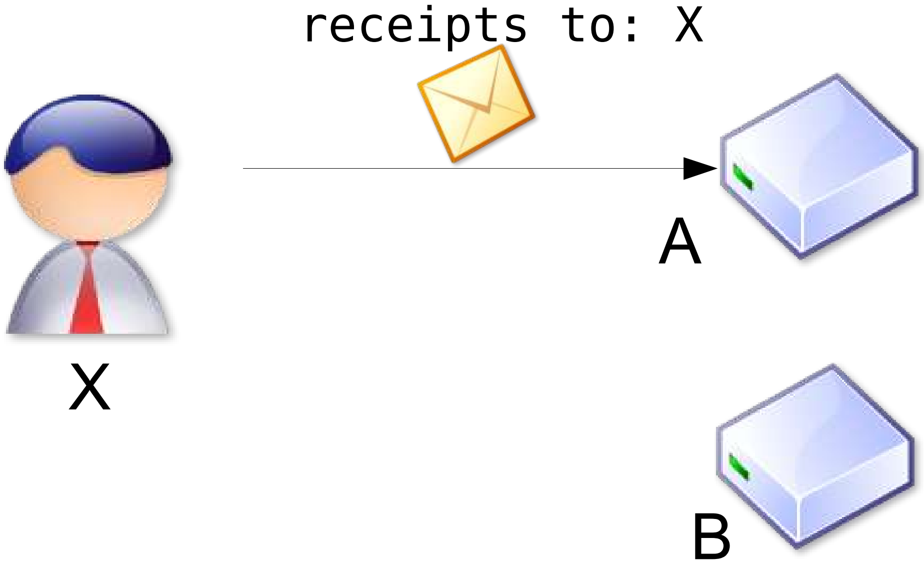
Mail List Management - Mail Loops (Example)



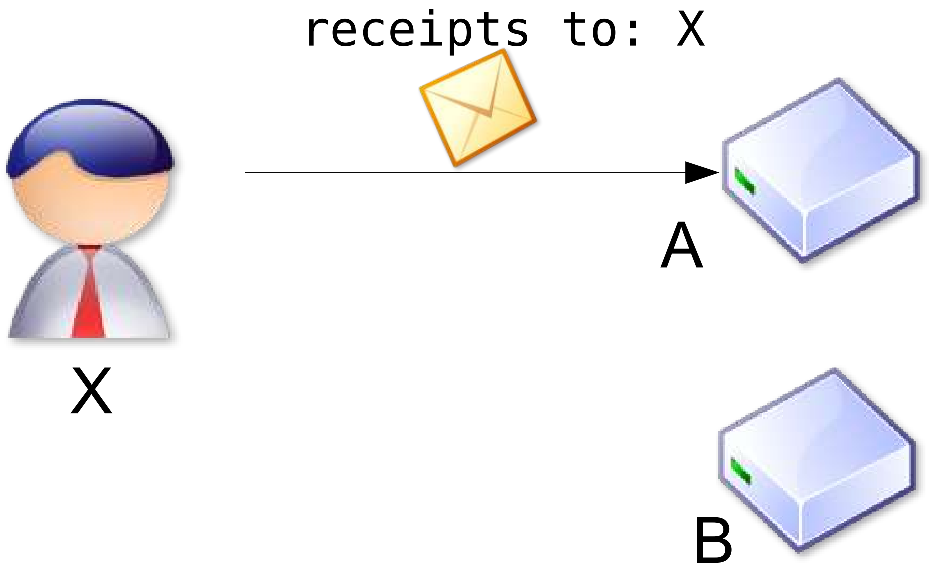
Mail List Management - Receipts

- Mail List Agent Signed Receipt Policy Processing
  - A MLA often needs to propagate forward the receipt policy
  - Any MLA adds *"insteadOf"*, *"inAdditionTo"*, *"none"* to the history
  - Only last recipient needs to process
- No receipt, if originator has not requested
- If originator has requested, but MLA supersedes request: MLA may inform the originator

Mail List Management - Receipts (Example)

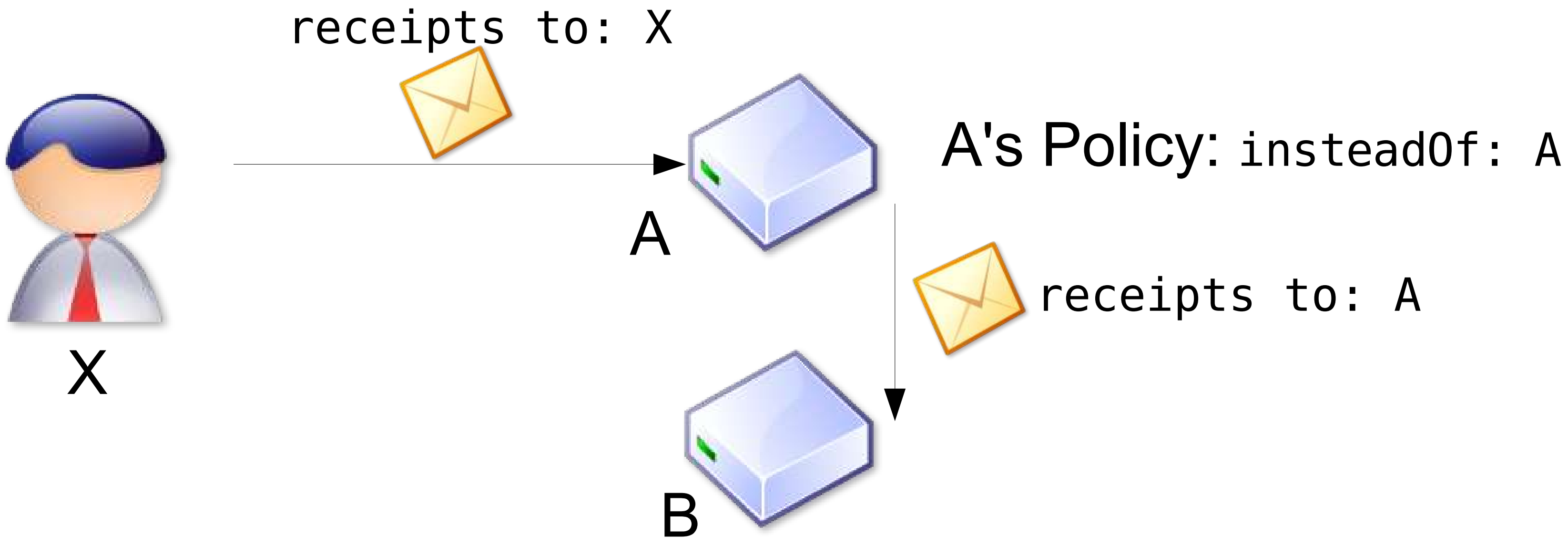


Mail List Management - Receipts (Example)



A's Policy: `insteadOf: A`

Mail List Management - Receipts (Example)



receipts to: X

A's Policy: `insteadOf: A`

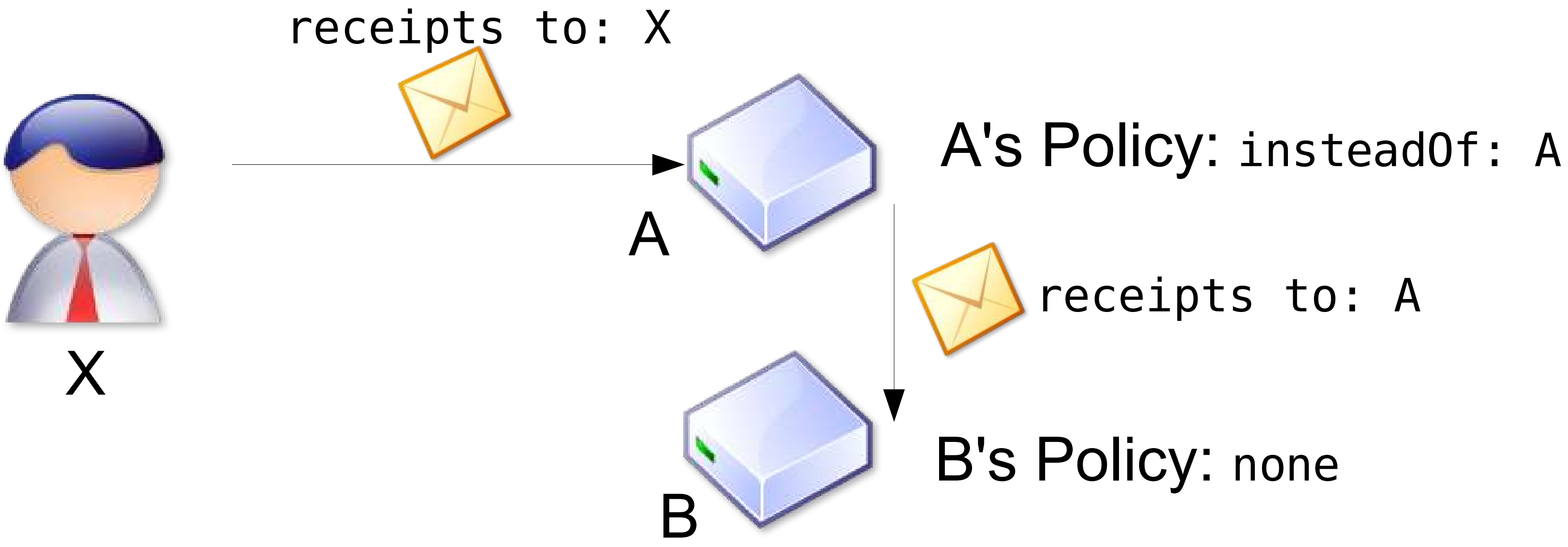
receipts to: A

X

A

B

Mail List Management - Receipts (Example)



receipts to: X

X

A

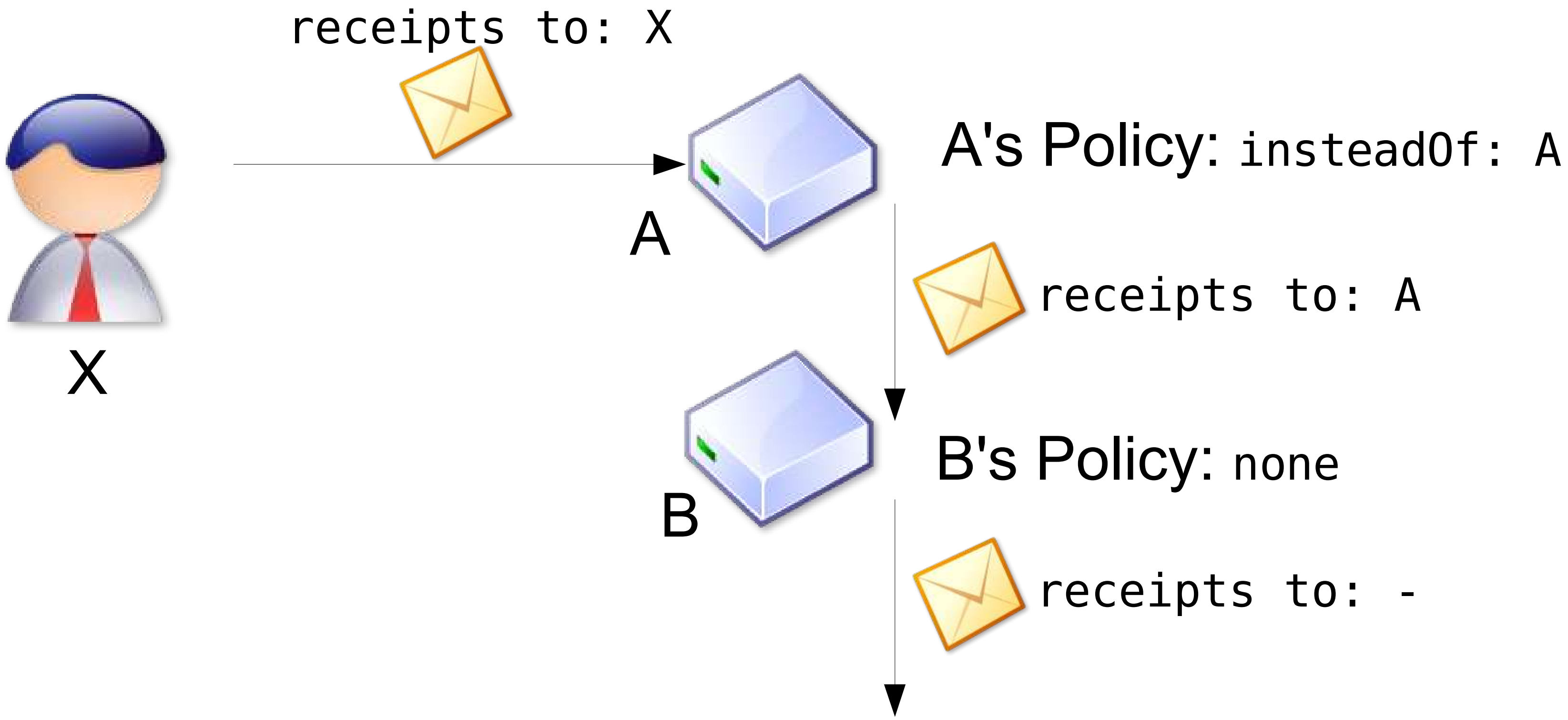
A's Policy: insteadOf: A

receipts to: A

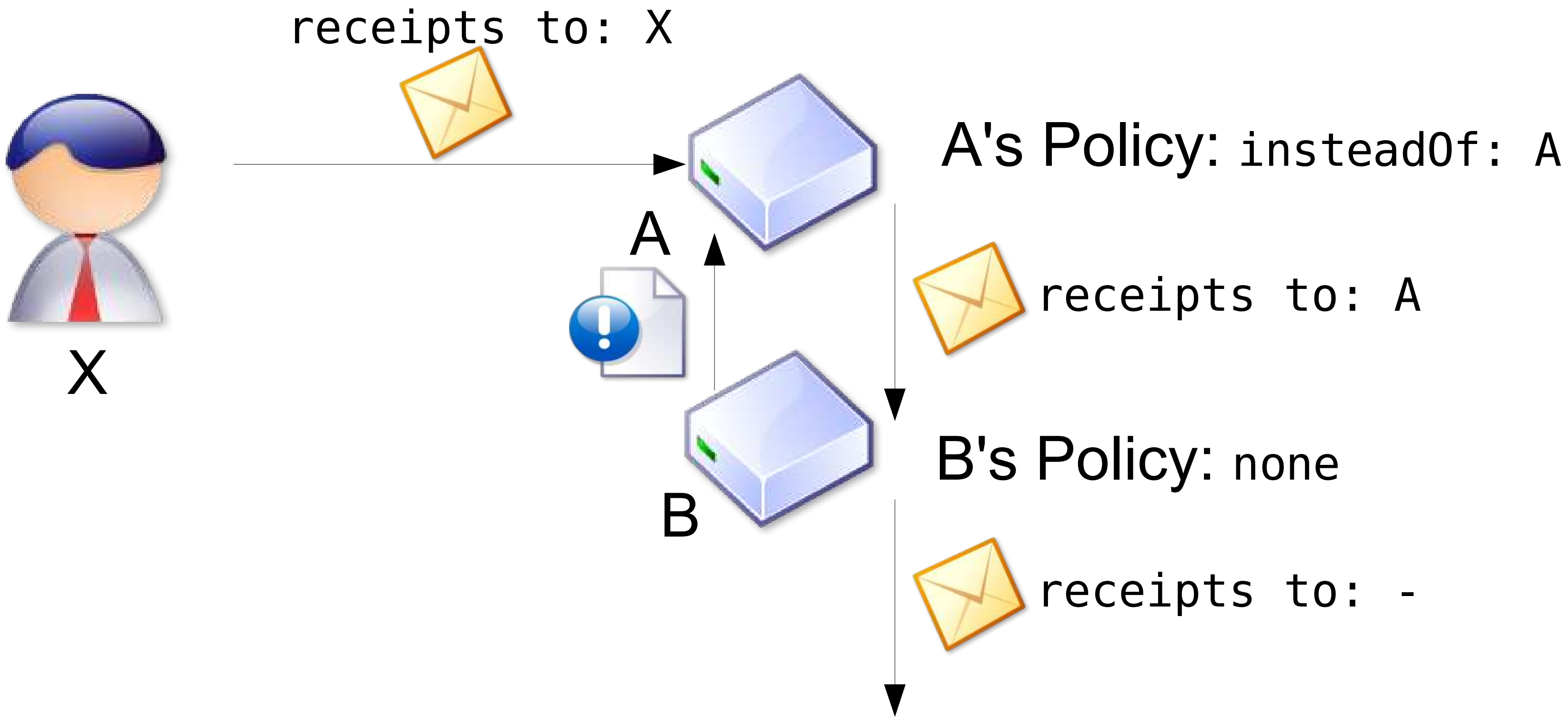
B

B's Policy: none

Mail List Management - Receipts (Example)



Mail List Management - Receipts (Example)



6 Signed Certificates

- Attacks
- Responses

Signing Certificate - Attacks

- Substitution Attack
  - Simple substitution of one certificate for a another
  - issuer and serial number in the SignerInfo is modified to refer to a new certificate
    - DoS-Attack where an invalid certificate is substituted for the valid
      => message is unverifiable, as the public key no longer matches the public key used to sign
    - Substitution of one valid certificate for the original valid certificate where the public keys match
      => Message is validated under different constraints the originator intended

Signing Certificate - Attacks (continued)

- Reissue of Certificate Attack
  - Attack deals with a certificate authority (CA) re-issuing the signing certificate
  - may become more frequent as CA reissue their own root certificates

- Duplicate CA Attack
  - Setting up a CA that attempts to duplicate an existing CA
  - Issue a new certificate with the same public keys as the signer used

Signing Certificate - Responses

- Substitution Response
  - DoS cannot be prevented
  - No way to automatically identify the attack because it is indistinguishable from a message corruption.
  - No practical way to prevent users from getting new certificates with the same public key.
- Reissue of Certificate Response
  - A CA should never reissue a certificate with different attributes
- Duplicate CA Response
  - Only way: Never trust a duplicate CA

7 Conclusion

- Security Considerations

Security Considerations

- Mailing lists
  - Mailing lists that encrypt their content my be targets for DoS-Attacks if they to not prevent Mail-Loops. Using simple RFC822-Header spoofing it is easy to subscribe on encrypted mailing list to another, thereby setting up an infinity loop.
  - Ciphertext Attacks: MLAs should notify an admin if a large number of undecryptable messages are receives

Security Considerations (continued)

- Signed Receipts
  - Recipient must not send back a reply if it cannot validate the signature.
  - Senders should encrypt receipts to prevent a passive attacker from gleaning information
- Security Labels
  - Senders must not rely on recipients' processing software to correctly process security labels
    - some S/MIME clients may not understand security labels but display a labeled message
    - Error response sent to originator and that error bounces back => unlike that the bounce message will have a proper security label

Details: RFC 2634