

Mixtasy:

Remailing on Existing Infrastructure

Anonymized Email Communication Easily Deployable Using SMTP & OpenPGP

Master's thesis presentation
@Young Researchers' Day 2016
St Johann im Pongau (11.10.2016)

by Johannes Burk

1st Reviewer:

Dipl.-Inform. David Stezenbach

2nd Reviewer:

Priv.-Doz. Mag. DI. DI. Dr.techn. Karl Michael Göschka

Introduction

- Secure messaging is a big research area
- Plain email did not provide any security or privacy feature
 - But it's still heavily used
- TLS and openPGP, S/MIME isn't enough
 - Metadata still readable
- Eavesdroppers/Adversaries are everywhere (attention tinfoil hat carriers!)

Objective

- Build a secure and privacy preserving asynchronous messaging prototype solution
- ... With good adoption properties (design on top of existing infrastructure)

Main Parts

- Requirement Definition
- Technologies & Existing Work
- Design Considerations
- Protocol Specification (wire protocol)
- Protocol Implementation (tool, prototype)

Requirements: Security and Privacy

Security

- Confidentiality, integrity and authenticity
 - end-to-end
- Anonymity Preserving
 - conversation security feature must not break transport privacy

Privacy

- Participation Anonymity & Global Adversary Resistance
- Unlinkability
- Sender Anonymity

Requirements: Usability and Adoption

Usability

- Keep Email Properties
 - asynchronicity
 - message drops/delays
- Easy Initialization

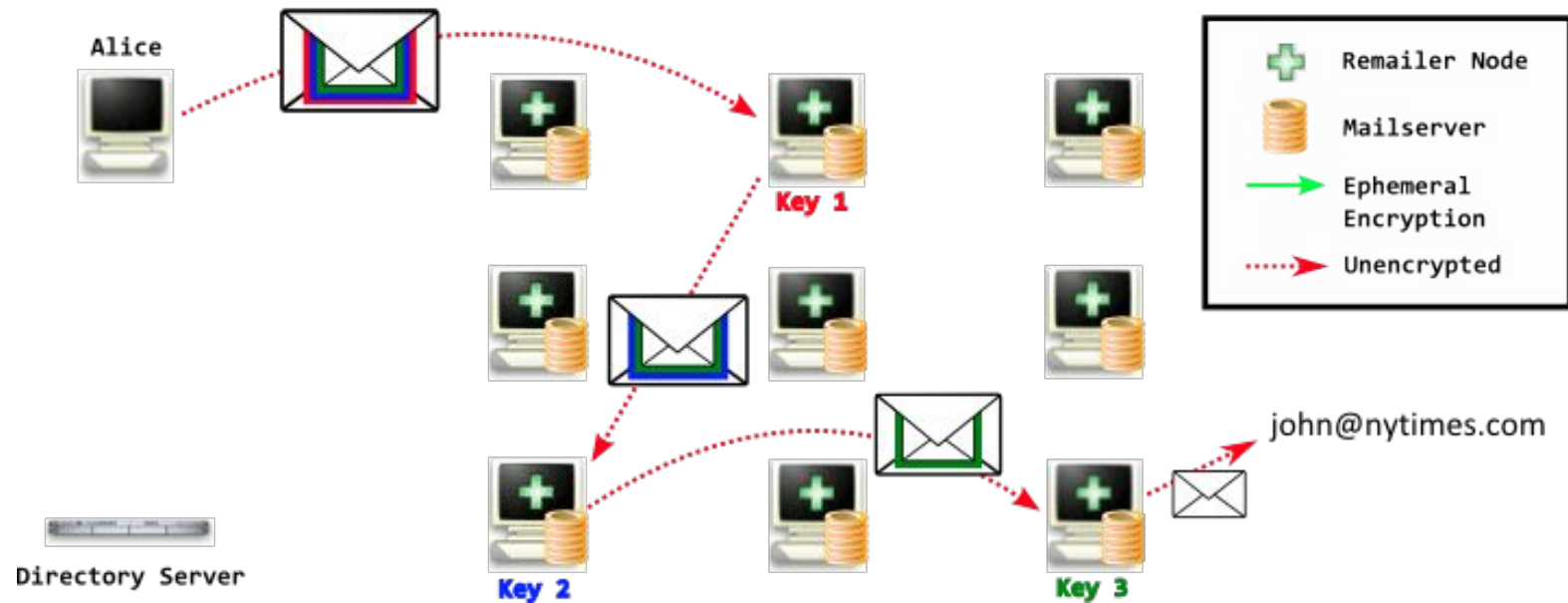
Adoption

- Compatibility to existing Infrastructure
- No Additional Service
- Scalable

Existing Work: Remailer

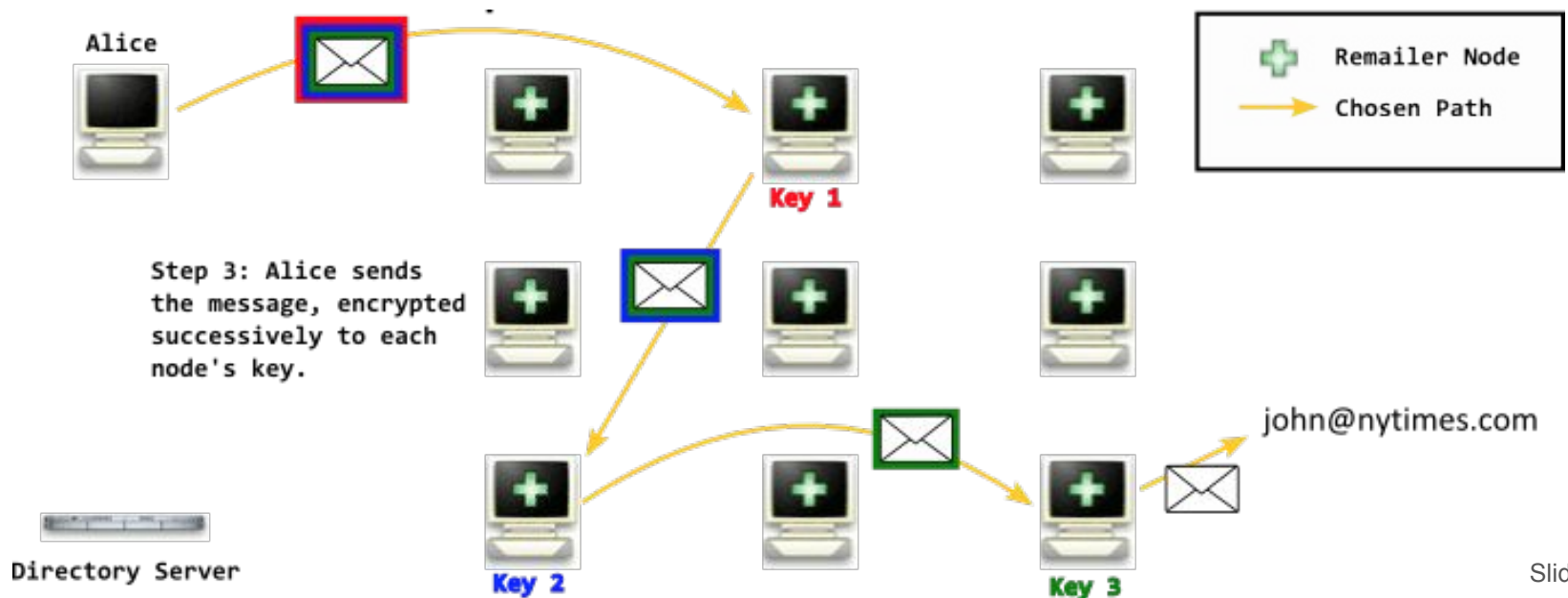
- Based on mix networks
- Different types (evolution caused)
 - Type 0: Pseudonymous/Nym remailer
 - Just for pseudonymization
 - Type 1: Cypherpunk
 - Encryption not mandatory
 - Type 2: Mixmaster
 - Outdated crypto (RSA-1024, (3DES), MD5, ...)
 - Type 3: Mixminion
 - Doesn't support SMTP

Mixmaster Remailer



The Idea

- Transport: SMTP
- Encryption: openPGP (GnuPG)
- Directory Service: openPGP Key Server (no additional service!)
- Implementation: Postfix Filter (adoption!) + client for sending mails



Design Considerations I

- Encryption: confidentiality & prevent tracking by content
 - layered encryption between sender and mixes/receiver
- Mixing Algorithm: blur the trace of a message (anonymity)
 - Timed dynamic-pool mix
- Message Size: prevent tracking by size (anonymity)
 - Uniformed; repadding at each mix

Design Considerations II

- Replay Attack prevention (anonymity)
 - Cache message hashes
- Tagging attack prevention (anonymity)
 - Message data verification
- Dummy Traffic: complicate blending attacks & reduce message delays
 - inject dummy messages
- Abuse & Spam protection
 - cost based spam protection

Protocol Specification: Mixtasy

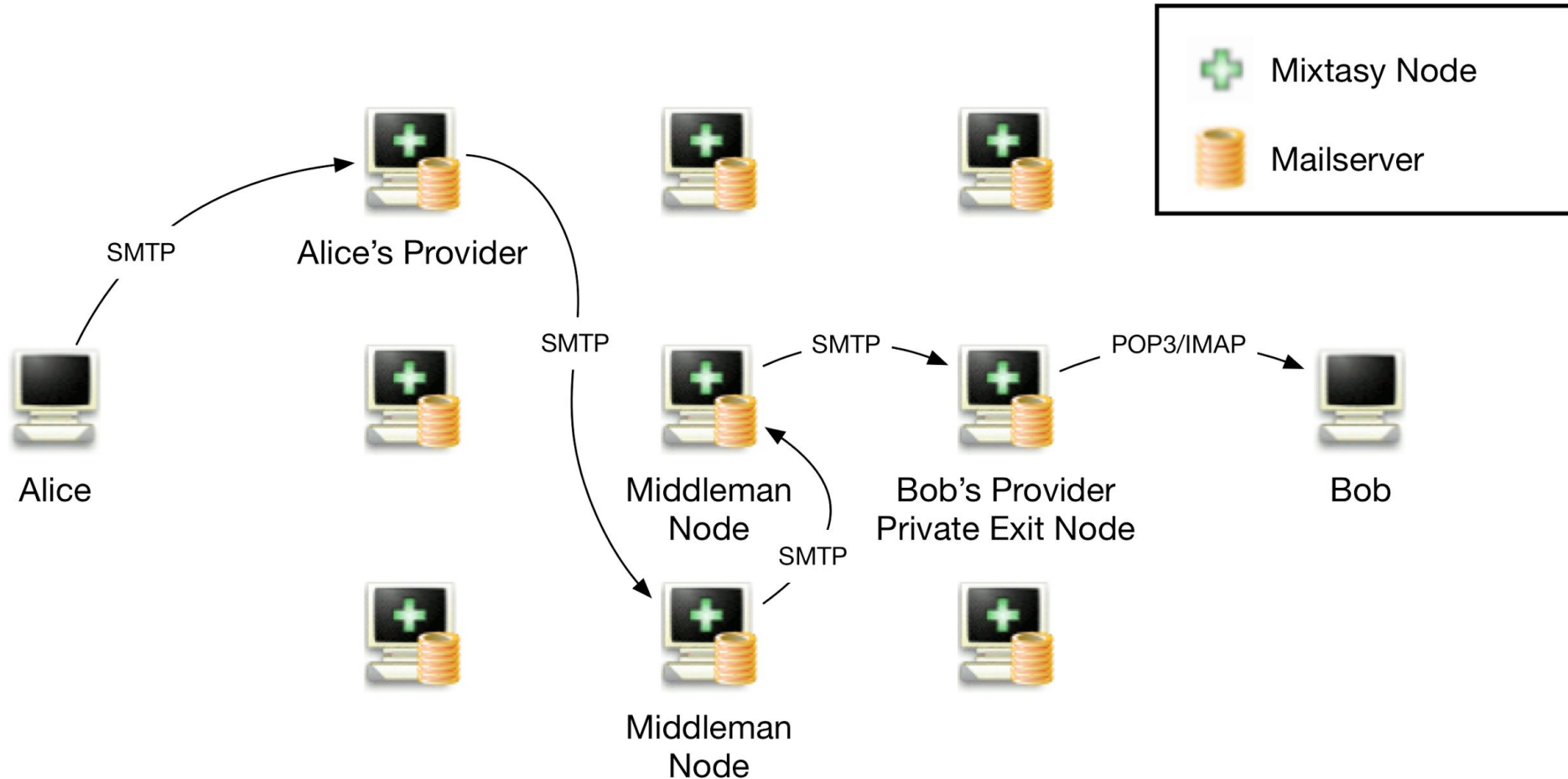
Reused Technologies

- SMTP [RFC-5321]
- Internet Message Format [RFC-5322]
- OpenPGP [RFC-4880]

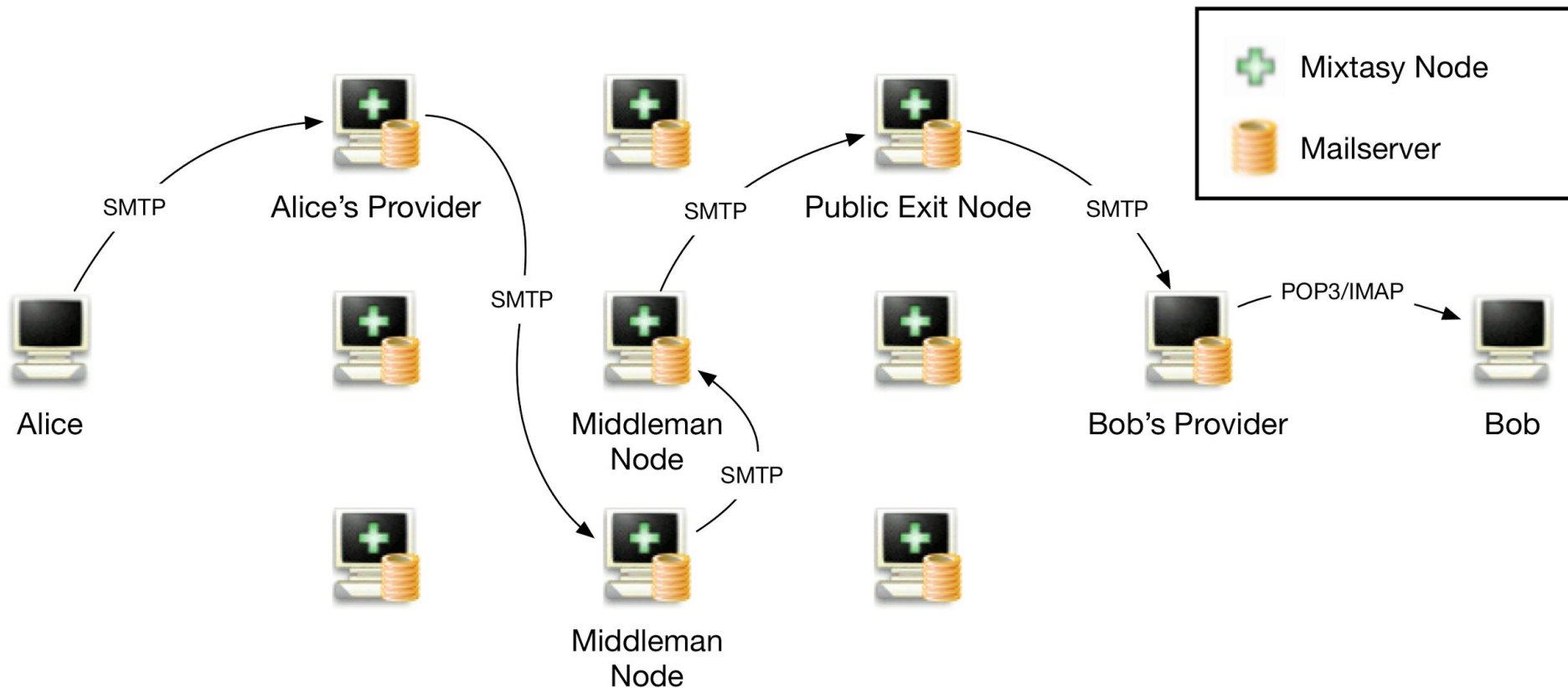
Overall Design

- Mix Network design
- OpenPGP encrypted emails (layered encryption → Mix Network)
- Transport with SMTP
- Sender needs Mixtasy software to create messages
- Mix Node: usual MTA with Mixtasy addon

Mixtasy Design: Receiver's Provider supports Mixtasy



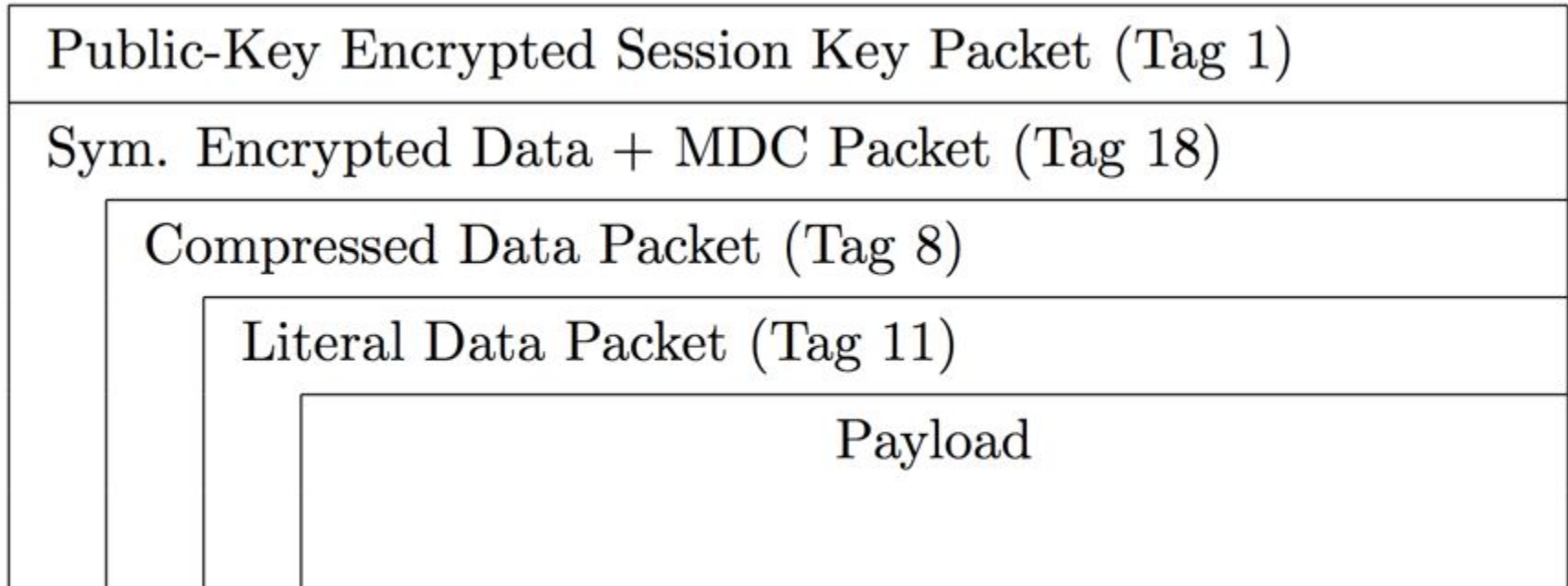
Mixtasy Design: Receiver's Provider doesn't support Mixtasy

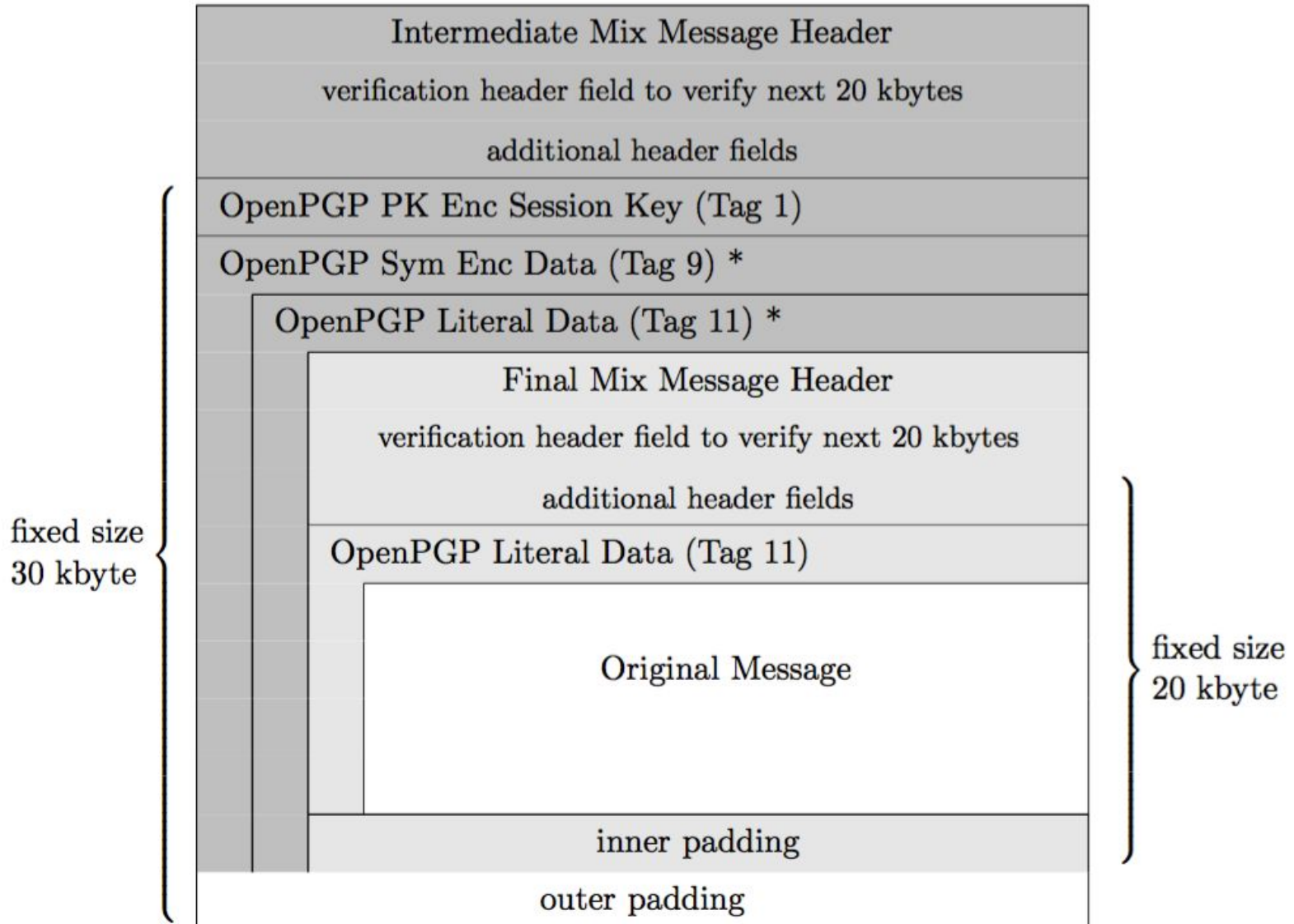


Message Format

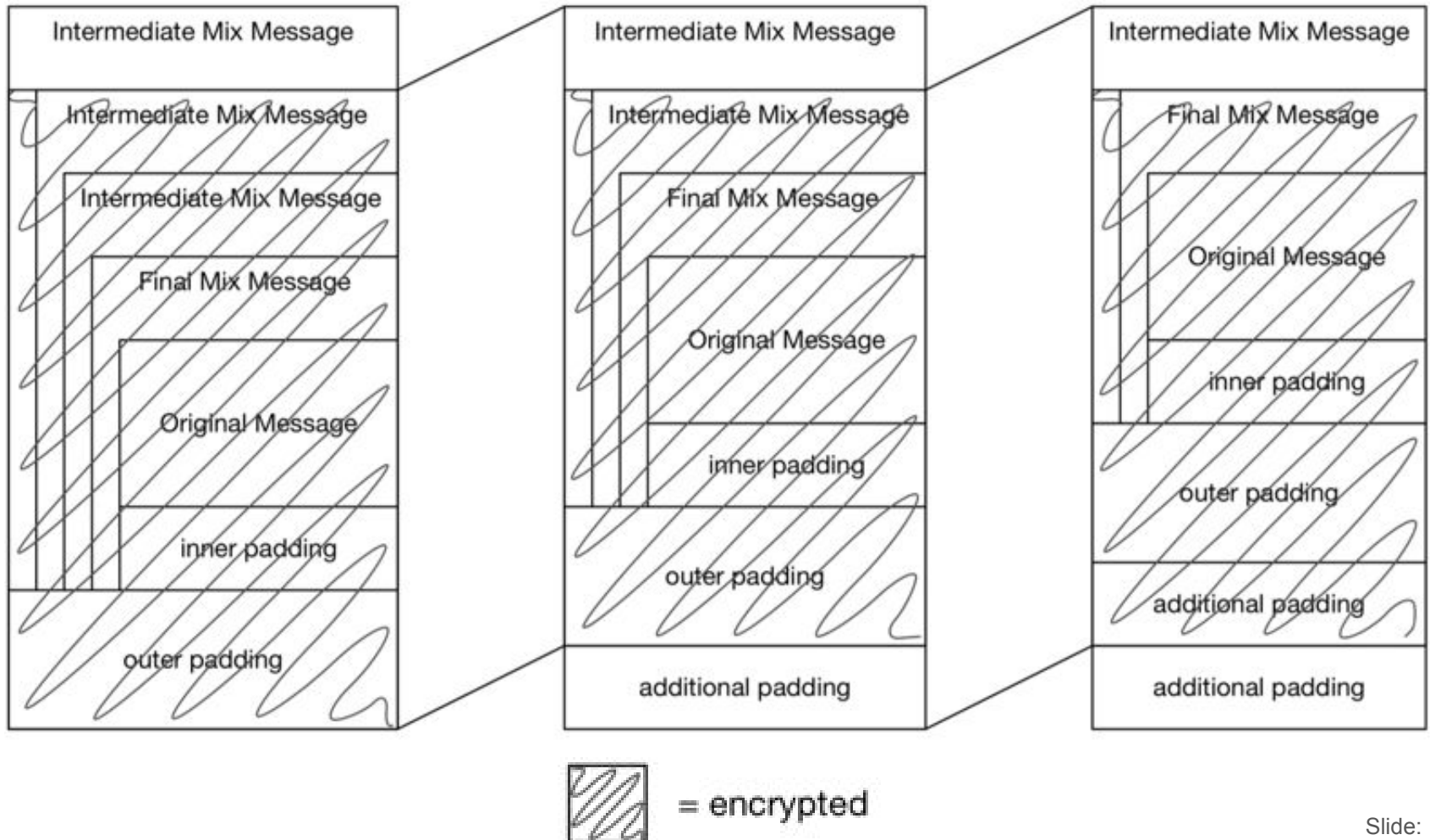
- Original Message
 - As composed by the sender
- Final Mix Message
 - Wraps an original message
- Intermediate Mix Message
 - Contains another intermediate or a final mix message

OpenPGP Message Format Example



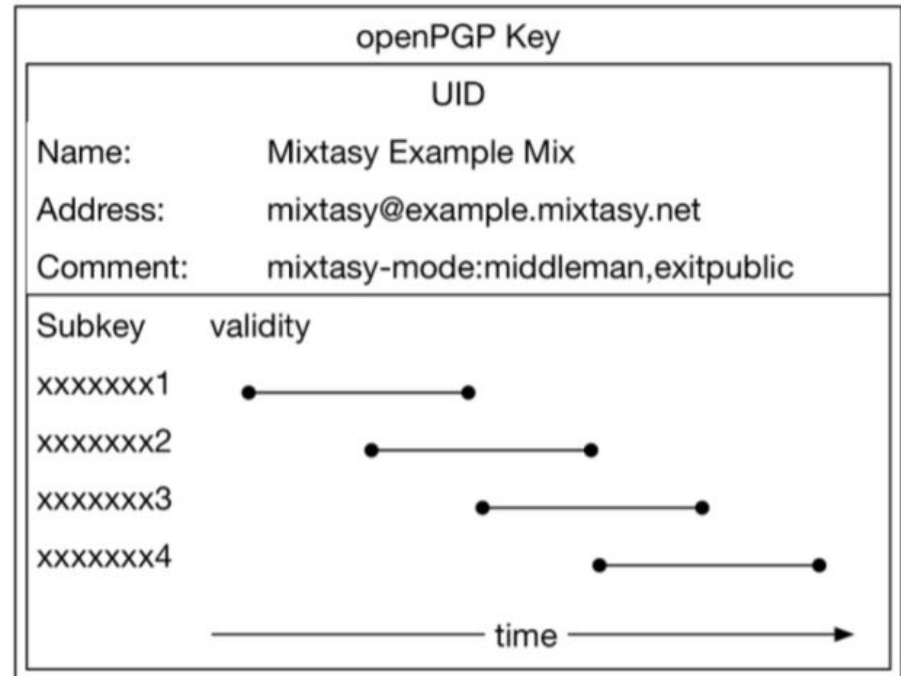


Decryption and re-padding



Keys

- Long-term OpenPGP key (trust establishment)
- Short-term sub-keys (encryption)
- Distribution over public PGP key server
- Discovery via search for
“mixtasy@”



Prototype

Available
on GitHub

- Written in Python, makes use of GnuPG
- CLI Client to create mails
 - Including: Mix discovery and key retrieval, Path selection, constructing single part messages, sending via SMTP
- Postfix Filter to operate a mix node
 - Including: Strip of encryption layer, Verification check, Re-padding to fixed message size
- Not implemented yet:
 - multi part and dummy messages, mixing algorithm, replay attack prevention

Conclusion

- Remailer protocol design and prototype created
 - Mostly specified by composing existing technologies
 - Deployable by upgrading existing MTAs
 - Receiver just needs OpenPGP software
- Future work
 - Implement full specification
 - Detailed evaluation/auditing
 - Research on dynamically change timed dynamic-pool mix parameters
 - Extend the protocol by an anonymous reply feature

Download Slides and Master's Thesis, Try out or Contribute

- <http://mixtasy.net/>
- <https://github.com/jojoob/mixtasy/>