Mixminion: Design of a Type III Anonymous Remailer Protocol

G. Danezis, R. Dingledine, N. Mathewson
IEEE S&P 2003

Presented by B. Choi in cs6461
Computer Science
Michigan Tech

Motivation

- Anonymous email only
  - High latency vs. near real-time (onion routing)
- Anonymous email implementations
  - Type 1: Cypherpunk (80's)
    - vulnerable to replay attacks
  - Type 2: Mixmaster(92)
    - message padding and pooling
  - Type 3: Mixminion (2003)
    - Anonymous Replies!

Reply block?

- Most or many systems support sender anonymity
- Pynchon Gate supports receiver anonymity in an interesting way (P2P file sharing: 2005)
  - Send everything to everywhere (everyone)
- Is receiver anonymity too hard to achieve?
  - First of all, receiver has to use pseudonyms
    - Pseudonym policy: how many, valid period, ...

Reply blocks

- Chaum(‘81), BABEL (‘96), Mixmaster (92) ..
  - Entire path is chosen by the sender
    - Variations are possible
    - BABEL RPI is invisible to passive external attackers
    - BABEL RPI is visible to internal passive attackers (mix)
  - Can be used multiple times?
    - Good for communication efficiency
    - Bad for anonymity due to potential path information leaking
    - Adversary could utilize the pattern of the same reply block

Fundamental solution to the reply block problem?

- One way is to use single-use reply blocks (SURB)
- Reply messages are indistinguishable from forward messages even to mix nodes
- Effect: both reply and forward messages share the same anonymity set
-

SURB

- How to design SURB?
  - Sender generates SURB
  - To defeat replay, each intermediate node has to maintain some information of each message it has processed (seen)
  - Message signature?
    - Lifetime of a message signature (ID)?
    - Min, hour, day, month ...
    - Mixminion: hashes of the headers and key rotation
  - Mixminion drops messages with duplicate headers!

Usage model

$$\alpha \rightarrow \mathrm{Nym} : \{\mathrm{Register}, \alpha, V_\alpha, \alpha_1 \ldots \alpha_n\}_{S_\alpha}$$

$$B \rightarrow \mathrm{Nym} : \alpha, M$$

$$\mathrm{Nym} \rightarrow \alpha_i : M$$

$$\alpha \rightarrow \mathrm{Nym} : \{\mathrm{Register}, \alpha, V_\alpha\}_{S_\alpha}$$

$$B \rightarrow \mathrm{Nym} : \alpha, M$$

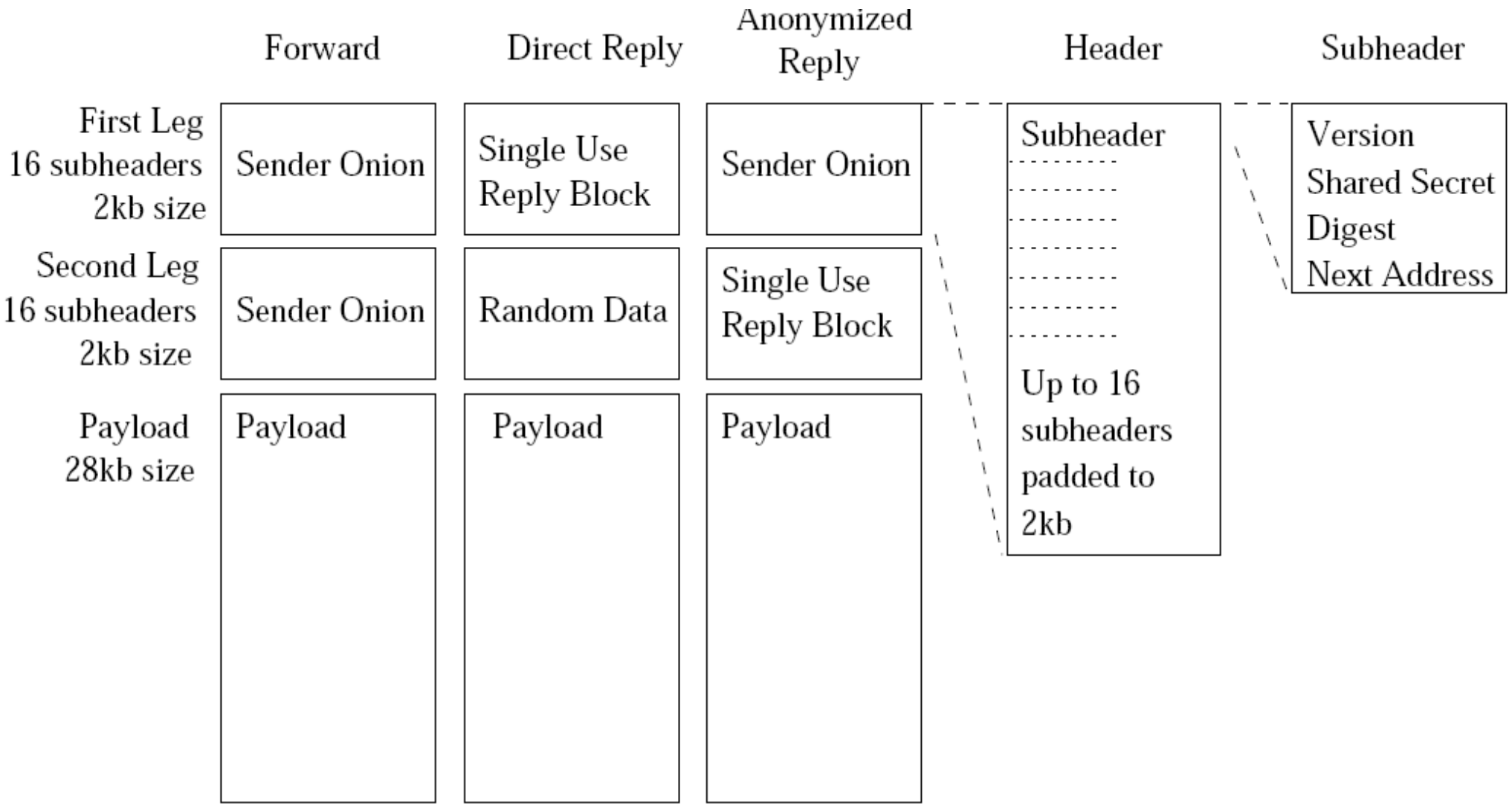
$$\alpha \rightarrow \mathrm{Nym} : \{\mathrm{Query}, \alpha, \alpha_1 \ldots \alpha_n\}_{S_\alpha}$$

$$\mathrm{Nym} \rightarrow \alpha_i : M$$

Usage model

- Directory servers
  - Not clear
- Membership protocols
  - Assumed
- Periodic key rotation
  - Not detailed

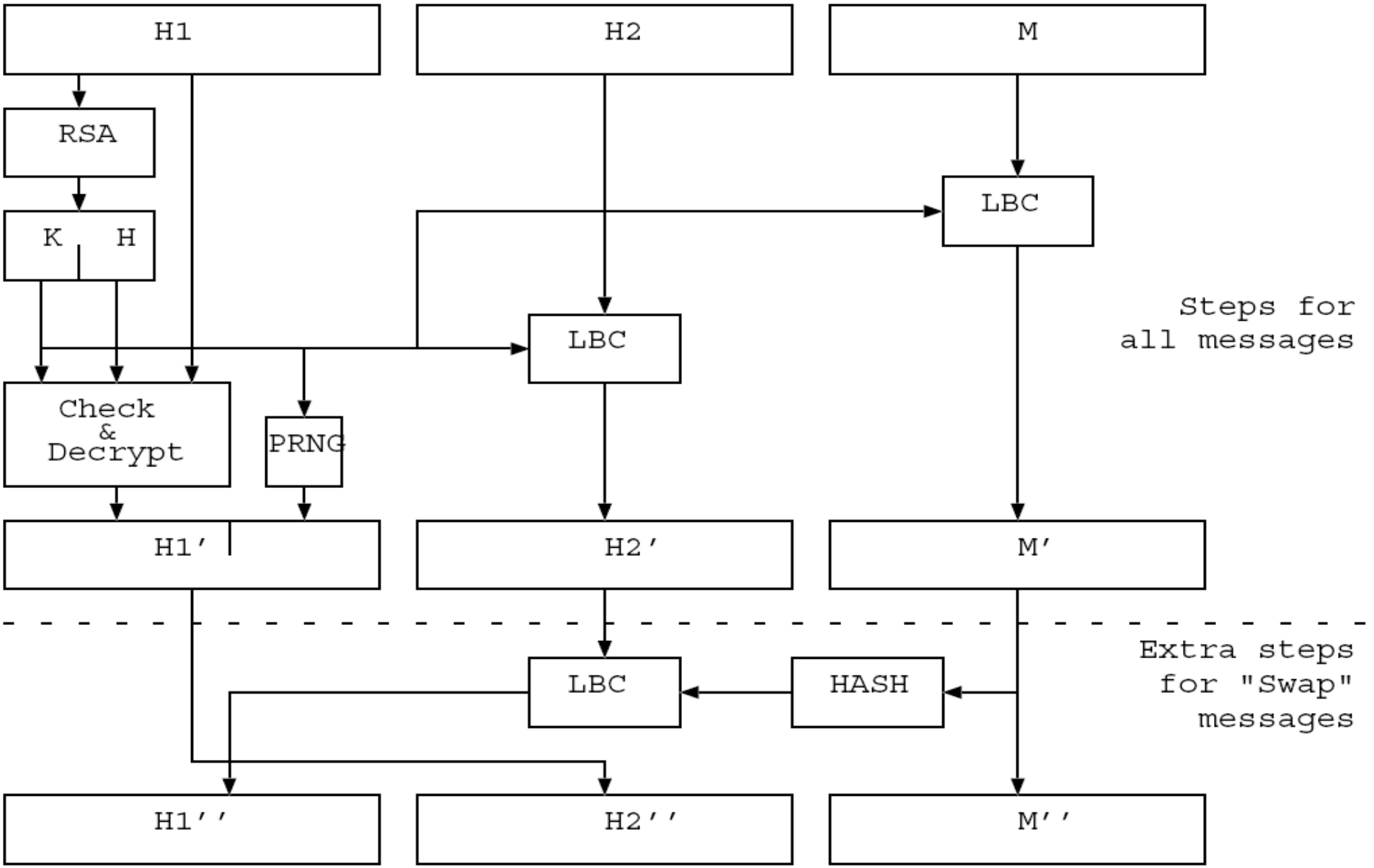
Header configurations



Tagging Attack

- SURB provides a better opportunity for the adversary to tag message to find out the path information
- Indistinguish-ability of forward and reply messages require something interesting
  - Separated encryption of header and payload
- Attacker tags part of the message payload and detects the mark at a later node in the path
  - Path is identified after repeated tagging attacks

Header Swap


H1
H2
M
RSA
LBC
K
H
LBC
Steps for
all messages
Check
&
Decrypt
PRNG
H1'
H2'
M'
LBC
HASH
Extra steps
for "Swap"
messages
H1''
H2''
M''

Effectiveness

- Analysis of tagging
  - Forward message
    - First leg, second leg
  - Direct reply message
    - Same to the conventional onions
  - Anonymized reply messages
    - Same as forward message