SureMail: Notification Overlay for Email Reliability

Sharad Agarwal & Venkat Padmanabhan

Microsoft Research

Dilip Antony Joseph

UC Berkeley

HotNets 2005

Motivation

- Silent email loss
  - email “vanishes” without sender/recipient knowledge
  - can be costly even if relatively rare
    - missed opportunities, misunderstanding, or worse
- Nontrivial problem
  - anecdotal evidence
  - measurement studies
    - 0.69% loss rate [Lang & Moors 2004]
    - 0.1-5% loss rate [Afergan & Beverly 2005]
  - commercial offerings to address the problem
    - e.g., Pivotal Veracity, Zenprise

HotNets e-ticket

"We have sent it through again. If you do not receive it with in an hour or two, please let us know."

Funding proposal

"No I never got and I never acked it... My last mail from you was on XYZ."

IMC 2005 decision notification

"I recd reviews for one paper (#X) but not that of #Y."

IMAP server upgrade problems

"Some unanticipated migration problems occurred that may have caused some lost or delayed email."

Motivation

- Silent email loss
  - email “vanishes” without sender/recipient knowledge
  - can be costly even if relatively rare
    - missed opportunities, misunderstanding, or worse
- Nontrivial problem
  - anecdotal evidence
  - measurement studies
    - 0.69% loss rate [Lang & Moors 2004]
    - 0.1-5% loss rate [Afergan & Beverly 2005]
  - commercial offerings to address the problem
    - e.g., Pivotal Veracity, Zenprise

Silent Email Loss

- Why email loss?
  - spam filtering: big problem ⇒ aggressive filtering
    - MS: 90% of emails discarded before hitting user mailboxes
    - AOL: 100 emails per month to maintain IP white-listing
  - server failures and upgrades
    - SMTP is not end-to-end reliable
- (Non-)Delivery status notifications
  - compounds spam problem
  - raises privacy concerns
- So email loss is often silent

Fixing the Problem

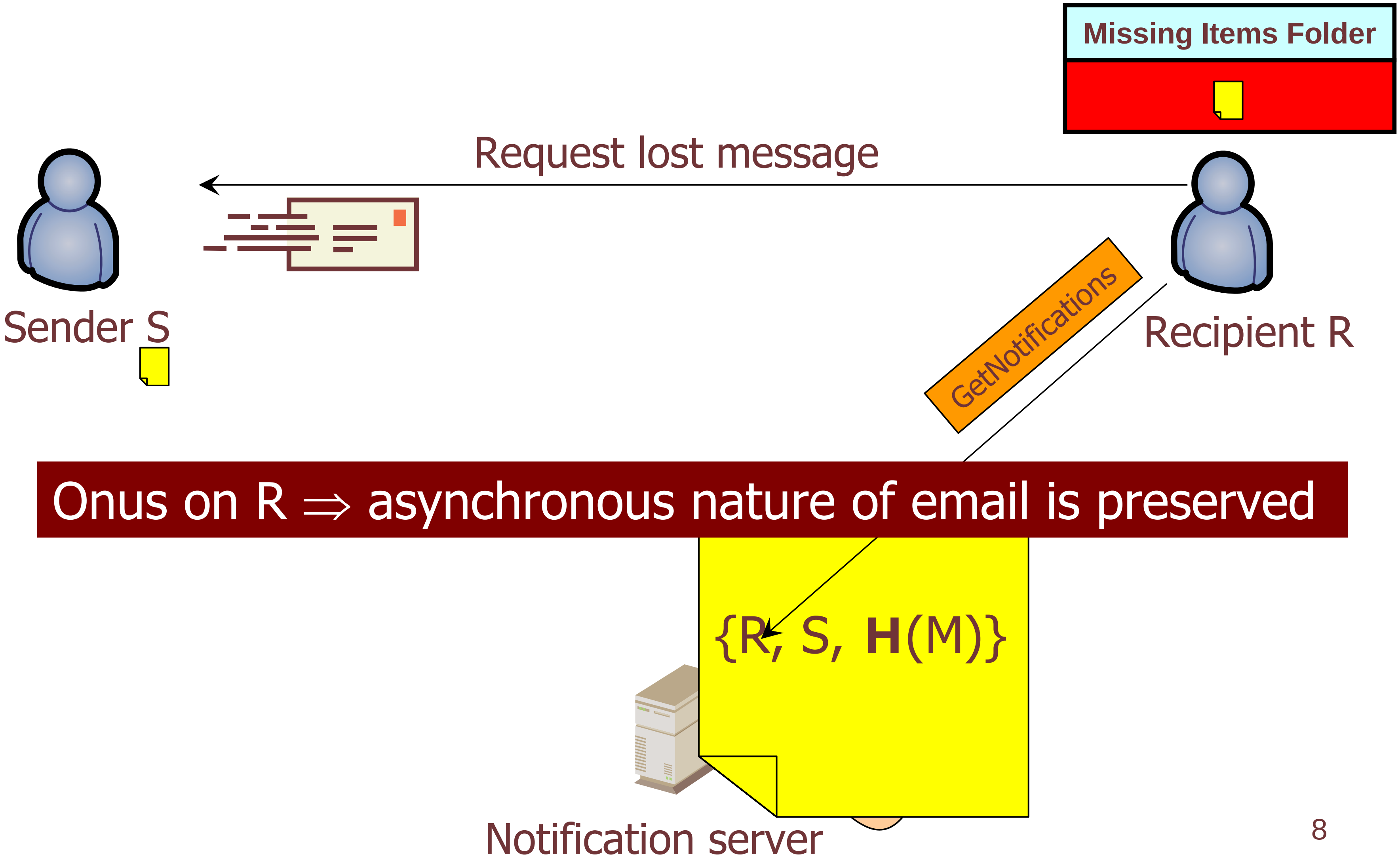
- Improve the email delivery infrastructure
  - more reliable servers
    - e.g., cluster-based (Porcupine [Saito ’00])
  - server-less systems
    - e.g., DHT-based (POST [Mislove ’03])
  - total switchover might be risky
- “Smarter” spam filtering
  - moving target ⇒ mistakes inevitable
  - non-content-based filtering still needed to cope with spam load

SureMail

- Address the problem from the outside
  - add separate notification overlay
  - emails & email delivery infrastructure left untouched
  - eases deployment, bounds the worst case

- Design requirements:
  - minimize demands on infrastructure and users
  - preserve asynchronous operation and privacy
  - maintain defenses against spam and viruses
  - minimize overhead

Basic Operation



Onus on R ⇒ asynchronous nature of email is preserved

Notification Overlay

- Decentralized
  - limited collusion among the constituent nodes

- Efficient notification server lookup
  - e.g., R → **H**(R) in a DHT setup

- Agnostic to actual implementation
  - end-host-based (e.g., always-on user desktops)
  - infrastructure-based (e.g., “NX servers”)

Challenges

- Privacy
  - information on users’ email habits or even just whether an email address is active could be leaked
- Notification spam
  - spammers could spoof notifications and burden users
  - “annoyance attacks” discredit notifications in general
- Even the notification infrastructure isn’t trusted
- No universal PKI for email users

SureMail Goals

- Protect the recipient’s identity
  - attacker shouldn’t be able to learn R’s identity or monitor the volume of notifications intended for R

- Protect the sender’s identity
  - attacker shouldn’t be able to learn S’s identity or monitor the volume of notifications posted by S

- Block notification spam
  - attacker shouldn’t be able to spoof notifications

{R, S, **H**(M)}

Assumptions

- No email eavesdroppers
  - bigger problems otherwise

- Limited collusion among notification nodes
  - needed only to avoid leaking information on whether or how many notifications R is receiving

Key Mechanisms

#1: Email-based handshake

#2: Decoupled registration and notification

#3: Email-based shared secret

#4: Reply-based shared secret

{**H**(R), S, **H**(M)}

#3: Email-based shared secret

Goal: prevent snooping on sender identity

💡 Email M_{old} from S to R in known only to S and R

- **H**(M_{old}) could serve as implicit identifier of S to R
- But it doesn’t quite serve as authenticator for S:
  - D_{not} knows **H**(M_{old}), so it could spoof notifications from S
  - even other attackers could do so by first sending $M_{spoofed}$ purporting to be from S

{**H**(R), **H**(M_{old}), **H**(M)}

#4: Reply-based shared secret

Goal: block spoofing of notifications

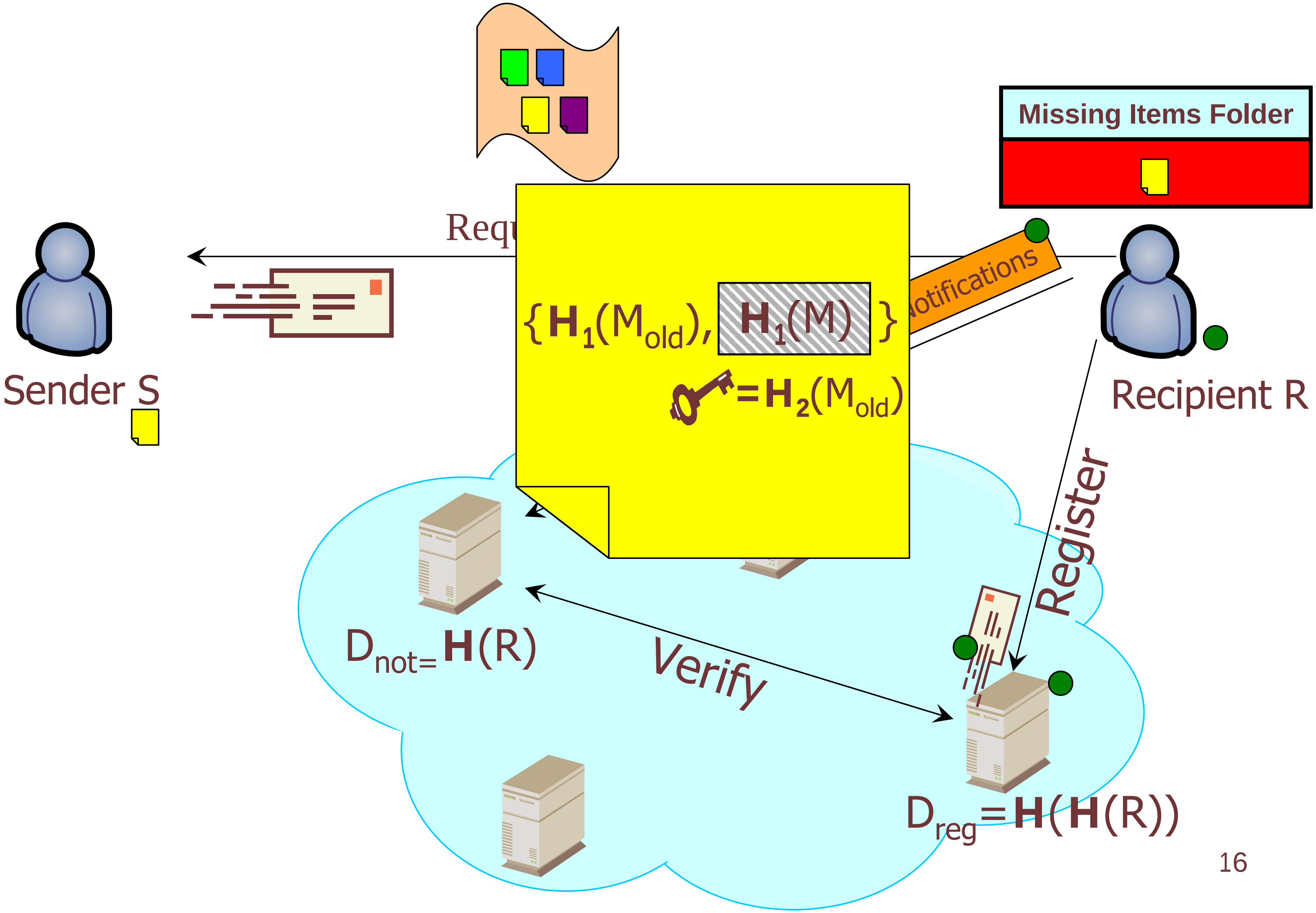
💡 Users rarely have conversations with spammers

- R remembers (hashes of) recent emails from S that it has replied to
- If S receives a reply to M_{old} it had sent R, M_{old} can serve as a shared secret between S and R
  - $\mathbf{H}_1(M_{old})$ as implicit identifier, $\mathbf{H}_2(M_{old})$ as authenticator
- Hard for a spammer (even D_{not}) to spoof

$$\{\mathbf{H}(R), \mathbf{H}_1(M_{old}), \boxed{\mathbf{H}_1(M)}\}$$

🔑$=\mathbf{H}_2(M_{old})$

Putting it all together



Other issues

- Reply-detection:
  - “in-reply-to” insufficient, indirect checks needed
- Reducing overhead:
  - look for implicit ACK (reply) or NACK (bounce-back)
  - post notifications selectively (for “important” emails)
- First-time “legitimate” senders:
  - indistinguishable from spammers
- Mobility:
  - reply-based scheme naturally lends itself to migration

Status

- Ongoing measurement experiment
- Design being refined
- Implementation in the works

Discussion

#1: Should the notification system be folded into the email infrastructure?

- Separation is advantageous:
  - provides failure independence
  - keeps the notification layer simple
    - small, fixed format notifications don’t require the same kind of processing as virus-laden email
    - no direct benefit for spammers
  - provides engineering convenience
    - fewer dependencies, easier deployment

Discussion

#2: Is there a social benefit to silent email loss because of the plausible deniability it provides?

- Any such benefit is far outweighed by the costs
  - Should cars be slightly unreliable because of the excuse it would give people when they miss an engagement?
- It is the asynchronous nature of email that is key and SureMail preserves that