Anonymous communication

- Bob and the server want to prevent outsiders from knowing they are communicating
  - Unlinkability
- Bob wants to prevent the server from knowing his identity
  - Source anonymity
Previous work: Chaum-Mix

- Standard model for anonymous routing:
  - Forward message through a static path of nodes \((P_1, \ldots, P_L)\)
  - Encrypt message \(M\) using public node keys in reverse order
Previous work: Chaum-Mix

- **Drawback**: path is fragile and hard to maintain
  - When any node/link fails, must rebuild entire path (expensive)
  - Source can not receive error messages, must use E2E timeouts

- **Drawback**: computationally expensive
  - Each message is encrypted with layers of asymmetric encryption
Other related work

• Chaum-Mix based
  – Onion routing [Syverson et. al 1997]
    • Pair-wise symmetric keys between nodes
  – Tarzan [Freedman et. al 2002]
    • Symmetric session keys and relay through nodes
  – Many other systems, e.g. Tor, etc.

• Probabilistic random walk
  – Crowds [Reiter et. al 1998]
    • No destination anonymity
    • Lower source anonymity [Diaz et. al 2002]

• Dining cryptographer network based
  – E.g. Herbivore [Sirer et. al 2004], P5 [Sherwood et. al 2001]
Cashmere overview

- Anonymous routing layer
  - Resilient to node churn, temporary node/link failures
    - Reduces path rebuild frequency
    - Result: much more stable paths
- Use structured overlays for group maintenance and inter-relay routing
- Comparable anonymity to Chaum-Mix
- Reduced vulnerability to predecessor attack [Wright et. al 2003 & 2004]
Outline

• Background & previous work
• Cashmere design
• Evaluation
• Summary
Design: use relay groups

- Instead of single nodes, use groups to relay traffic
- Relay functions if at least one member is reachable
- Leverage structured overlays (prefix based)
  - Relay group membership maintenance
  - Inter-relay routing
Relay group membership

- Each node assigned a nodeID
  - Assigned by a CA
  - Selected uniformly at random

- A relay group is a set of nodes sharing a common prefix
  - $\text{groupID} \equiv$ the shared prefix

- For example (Network size: $N$)
  - Relay group “$0XXXXX$”
  - Group size $\approx \frac{N}{2}$
Each node assigned a nodeID
- Selected uniformly at random

A relay group is a set of nodes sharing a common prefix
- $\text{groupID} \equiv$ the shared prefix

For example (Network size: $N$)
- Relay group “00XXX”
- Group size $\approx N/4$

Nodes estimate $N$ locally
- Routing table depth
- Source decides relay group size per session
Inter-relay routing

- Select a set of relay groups
  - Destination is member of a relay group
- Route message along the sequence of prefixes
  - 001XX → 100XX → 101XX → 010XX
- First relay member to receive the message is “root”
  - Broadcast to group members
  - Route to next relay group
- B receives broadcast message
Summary

- Benefits from structured overlay
  - Relay group maintenance
  - Inter-relay routing
  - Group broadcast
  - Locality-aware overlay routing
- No extra routing state per node
Prefix keys for relay groups

- Based on prefix, each relay group has key pair $K_{pub}, K_{priv}$
  - Each member uses $K_{priv}$ for group decryption
- Each node keeps key pairs for prefixes it shares
  - E.g. 12345 keys: 1XXXX, 12XXX, 123XX, 1234X, 12345
  - Retrieve from offline CA during ID assignment
- Store list of public keys for random prefixes
  - Obtained from trusted offline CA
Decoupling path and payload

- Chaum-Mix
  - Path embedded in encrypted layers around each payload
  - $L$ relays $\Rightarrow$ $L$ asymmetric operations at source and relay
Decoupling path and payload

- Cashmere
  - Decouple path and payload components
  - Path component: layered using asymmetric encryption
    - \( P_x \): prefix identifier for next hop
  - Payload component: symmetric encrypted layers w/ random keys
    - \( R_x \): random key
    - Symmetric encryption ensures message modified per hop
  - Path fixed per session (cacheable), payload changes per message
- Further extension: establish symmetric session key
  - All payload encrypted using symmetric key
  - See paper for further details

\[
\text{Path} = \langle P_{i+1}, R_i, P_i, R_{i-1}, P_{i-1}, R_{i-2} \rangle
\]

\[
\text{Payload} = \langle M \rangle_{\text{dest pub}}
\]

PubKey(P)

\( R_{i-1} \)
\( R_{i-2} \)
Message replies in Cashmere

• Destination replies without sacrificing source anonymity
  – Source generates random return path
    • Return path independent from forwarding path
    – Embed return path in original payload
    – Destination can send arbitrary reply message
  • Decoupling path and payload enables this
    – Further details in paper
Outline

• Background & previous work
• Cashmere design
• Evaluation
• Summary
Experiment setup

• Simulation
  – Analysis performed on random generated paths
  – Network size: $2^{14}$ (16K)
  – Prefix length: 12 bits
  – All attackers collude with zero latency

• Evaluation on PlanetLab
  – Implemented on FreePastry, (with RSA and Blowfish)
  – 128 Cashmere nodes
    • 32 machines geographically distributed over USA
    • 4 virtual nodes per machine
    • Four relay groups of size 4
Unlinkability

Anonymity using entropy metric [Diaz et. al 2002]
Resilience: expected path lifetime

• Churn
  – Exponentially distributed session times
    • median session time = 60 mins
  – Rate of node joins and failures is identical
  – Expected Cashmere path lifetime
    • Over one order of magnitude longer than node-based path
Path resilience based on Kazaa dataset

- Real distribution of Kazaa download time from [Gummadi et al. 2003]
- Reduce number of path rebuilds also reduce vulnerability to predecessor attack [Wright et. al 2003 & 2004]

NSDI, May 2005
Evaluation on PlanetLab

![Graph showing relative delay penalty vs. IP latency for Pastry and Cashmere protocols.]

- Y-axis: Relative Delay Penalty
- X-axis: IP Latency (ms)
- Two lines: Pastry (blue) and Cashmere (magenta)

NSDI, May 2005
Conclusion and future work

• Flexible and resilient anonymous routing
  – Relay messages through groups of nodes
  – Leverages structured overlay networks
  – Performance overhead is reasonable under churn

• Ongoing work
  – Scalable public key distribution
    • Leverage Identity-based encryption [Boneh et. al 2003]
  – Extending anonymous routing to multicast

http://www.cs.ucsb.edu/~ravenben/cashmere
Thank you!