Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms

EJ Jung
Goals

1. Hide what you wrote
   - encryption of any kind
   - symmetric/asymmetric/stream

2. Hide to whom you sent and when
   - pseudonym? proxy?
   - traffic analysis problem

3. Still receive a reply
   - hidden return address
Despite..

- No trusted authority
  - cannot send the mail to this and ask to forward

- Insecure underlying communication
  - cannot send the mail over “hot channel”
  - attacker can eavesdrop any message on any link
  - attacker can inject/modify/record any messages
Given: Everybody knows Bob’s public key

Only Bob knows the corresponding private key

Assumptions: 1. Attacker cannot guess the private key based on public key
               2. Attacker cannot convince Alice a wrong public key of Bob
                  - How to achieve this in real world?
Basic Mix Design

Mix

{r_1, {r_0, M}_{pk(B)}, B}_{pk(mix)}

{r_2, {r_3, M'}_{pk(E)}, E}_{pk(mix)}

{r_3, {r_5, M''}_{pk(B)}, B}_{pk(mix)}

{r_4, {r_5, M''}_{pk(B)}, B}_{pk(mix)}

{r_0, M}_{pk(B)}, B

{r_5, M''}_{pk(B)}, B

{r_3, M'}_{pk(E)}, E

Adversary knows all senders and all receivers, but cannot link a sent message with a received message.
Anonymous Return Address (0)

What's wrong with this?
- B knows who A is!
Anonymous Return Address (1)

message includes $K$ where $K$ is a fresh public key

$$\{r_1, K, \{r_0, K, M\}_{pk(B)}, B\}_{pk(mix)}$$

Response MIX

$$\{K, \{r_2, M'\}_K\}_{pk(mix)}$$

what's wrong with this??
MIX knows that $A = K$ (traceable)
Anonymous Return Address (2)

Q: Why A needs to encrypt \( \{K_1, A\}_{pk(mix)} \), not B?

M includes \( \{K_1, A\}_{pk(mix)} \), \( K_2 \) where \( K_2 \) is a fresh public key

\[ \{r_1, \{r_0, M\}_{pk(B)} , B\}_{pk(mix)} \]

\[ \{r_0, M\}_{pk(B)} , B \]

\[ A, \{\{r_2, M'\}_{K_2}\}_{K_1} \]

\[ \{K_1, A\}_{pk(mix)}, \{r_2, M'\}_{K_2} \]

Secrecy without authentication (good for an online confession service 😊)
Mix Cascade

- Messages are sent through a sequence of mixes
  - Can also form an arbitrary network of mixes ("mixnet")
- Some of the mixes may be controlled by attacker, but even a single good mix guarantees anonymity
- Pad and buffer traffic to foil correlation attacks
Small tricks

- Size-based correlation
  - send in fixed size blocks

- Timing-based correlation
  - send a random string even in idle times

- Frequency-based correlation
  - send always at maximum rate
Disadvantages of Basic Mixnets

- Public-key encryption and decryption at each mix are computationally expensive
- Basic mixnets have high latency
  - Ok for email, not Ok for anonymous Web browsing
- Challenge: low-latency anonymity network
  - Use public-key cryptography to establish a “circuit” with pairwise symmetric keys between hops on the circuit
  - Then use symmetric decryption and re-encryption to move data messages along the established circuits
  - Each node behaves like a mix; anonymity is preserved even if some nodes are compromised
Another Idea: Randomized Routing

- Hide message source by routing it randomly
  - Popular technique: Crowds, Freenet, Onion routing
- Routers don’t know for sure if the apparent source of a message is the true sender or another router
Sender chooses a random sequence of routers

Some routers are honest, some controlled by attacker
Sender controls the length of the path
Route Establishment

Routing info for each link encrypted with router’s public key
Each router learns only the identity of the next router
Location Hidden Servers

- Goal: deploy a server on the Internet that anyone can connect to without knowing where it is or who runs it
- Accessible from anywhere
- Resistant to censorship
- Can survive full-blown DoS attack
- Resistant to physical attack
  - Can’t find the physical server!
Creating a Location Hidden Server

Server creates onion routes to “introduction points”

Server gives intro points’ descriptors and addresses to service lookup directory

Client obtains service descriptor and intro point address from directory
Using a Location Hidden Server

- Client creates onion route to a “rendezvous point”
- Rendezvous point mates the circuits from client & server
- If server chooses to talk to client, connect to rendezvous point

- Client sends address of the rendezvous point and any authorization, if needed, to server through intro point

Diagram:
- Client (Alice)
- Rendezvous Point
- Server (Bob)
- Introduction Points
Deployed Anonymity Systems

- Free Haven project has an excellent bibliography on anonymity
  - [http://freehaven.net/anonbib/date.html](http://freehaven.net/anonbib/date.html)
  - Overlay circuit-based anonymity network
  - Best for low-latency applications such as anonymous Web browsing
- Mixminion ([http://www.mixminion.net](http://www.mixminion.net))
  - Network of mixes
  - Designed for high-latency applications such as anonymous email