Lecture 8: Privacy and Anonymity
Using Anonymizing Networks

CS 336/536: Computer Network Security
Fall 2015

Nitesh Saxena

Some slides borrowed from Philippe Golle, Markus Jacobson

Course Admin

• HW/Lab 3 posted
  – Covers Lecture 7 (SSL/TLS)
  – Due Nov 16
• Labs active this Friday
• Questions?
A Case History: AOL Web Search Query Log Leakage

• **AOL's disturbing glimpse into users' lives**, CNET News, August 7, 2006
  – 21 million search queries posed over a 3-month long period; 650,000 users
  – No user identification information released per se, **but??**
• Search Log still available:

Other Scenarios where privacy is important

• Location-based search
• Web browsing
• Electronic voting
• Electronic payments
• Email conversation
• ...
Today’s Outline

- Anonymizing Network (or Mix Network)
- Anonymizing Network Applications
- Requirements
- Robustness
- Types of Anonymizing Networks
  - Decryption based (Onion Routing)
  - Re-encryption based

Definition: Mix Server (or Relay)

- A mix server:
  - Receives inputs
  - Produces “related” outputs
  - The relationship between inputs and outputs is secret
Definition: Mix Network

- **Mix network**
  A group of mix servers that operate sequentially.

```
Server 1  Server 2  Server 3
```

Inputs  Outputs

Applications

- **Hide:**
  - “who voted for whom?”
  - “who paid whom?”
  - “who communicated with whom?”
  - “what is the source of a message?”

- Good for protecting privacy for
  - election and communication

- Used as a privacy building block
Electronic Voting Demonstration

1. “Who do you like best?”
   - Washington
   - Lincoln
   - Roosevelt

2. Put your ballot into a WHITE envelope and put again in a RED one and sign on it

   Jerry

Electronic Voting Demo. (Cont’d)

Administrators will

1. Verify signatures together
2. 1\textsuperscript{st} Admin. shuffles and opens RED envelopes
3. Send them to 2\textsuperscript{nd} Admin.
4. 2\textsuperscript{nd} Admin. shuffles again and opens WHITE envelopes
5. Count ballots together
A real system for elections

\[ \text{Sign}_{\text{voter } 1} (\text{encr}(\text{encr} (\text{vote}_1))) \]
\[ \text{Sign}_{\text{voter } 2} (\text{encr}(\text{encr} (\text{vote}_2))) \]
\[ \vdots \]
\[ \text{Sign}_{\text{voter } n} (\text{encr}(\text{encr} (\text{vote}_n))) \]

Mix Net \rightarrow \{ \text{vote}_1, \text{vote}_2, \text{vote}_3, \ldots, \text{vote}_n \}

Electronic Payment Demo.

- “Choose one person you like to pay $5”

\[ \text{Name of the person} \ (\underline{\text{__________}}) \]

- Put your ballot into an **WHITE** envelope and put again in a **RED** one and sign on it
Electronic Payment Demo. (Cont’d)

Administrators will
1. Verify signatures together
2. Deduct $5 from each account
3. 1st Admin. shuffles and opens **RED** envelopes
4. Send them to 2nd Admin.
5. 2nd Admin. shuffles again and opens **WHITE** envelopes
6. Credit $5 to recipients

For Payments

Sign_{payer_1} (encr(encr (payee_1))))
Sign_{payer_2} (encr(encr (payee_2))))
   .
   .
   .
Sign_{payer_n} (encr(encr (payee_n))))

D E D U C T

Mix Net

{payee_1}
{payee_2}
   .
   .
   {payee_n}

Credit

11/10/2015
Lecture 8: Privacy
For email communication

\[\text{encr (email}_1, \text{addressee}_1) \rightarrow \text{Mix Net} \rightarrow \ldots \rightarrow \text{Deliver}\]

\[\text{encr (email}_2, \text{addressee}_2) \rightarrow \ldots \rightarrow \ldots \rightarrow \text{Deliver}\]

\[\text{encr (email}_n, \text{addressee}_n) \rightarrow \ldots \rightarrow \ldots \rightarrow \text{Deliver}\]

To: Jerry
Don't forget to have lunch.

Other Uses

- Anonymous web browsing; web searching (Anonymizer)
Other Uses (Cont’d)

- Location privacy for cellular devices
  - Location-based service is GOOD!
    - Landline-phone calling to 911 in the US, 112 in Europe
    - All cellular carrier since December 2005
  - RISK!
    - Location-based spam
    - Harm to a reputation

Other Uses

- Anonymous VoIP calls
- Anonymous acquisition of security patches
Other uses (Cont’d)

Sometimes abuses

- Avoid legislation (e.g., piracy)
- P2P sharing of copyright content
- Terrorism: communication with media
  – 2008 Mumbai attacks

Requirements:
Privacy
Efficiency
Trust
Robustness
Requirements

1. Privacy
   Nobody knows who said what

2. Efficiency
   Mixing is efficient (= practically useful)

3. Trust
   How many entities do we have to trust?

4. Robustness
   Will replacement cheaters be caught? What if a certain number of mix servers fail?

But what about robustness?

I ignore his output
and produce my own

encr(Obama) → Hillary
encr(Obama) → Hillary
encr(Hillary) → Hillary

There is no robustness!
**Zoology of Mix Networks**

- **Decryption Mix Nets** [Cha81,…]:
  - Inputs: ciphertexts
  - Outputs: *decryption* of the inputs.

- **Re-encryption Mix Nets** [PIK93,…]:
  - Inputs: ciphertexts
  - Outputs: *re-encryption* of the inputs

---

**First Solution**

Chaum ’81, implemented by Syverson, Goldschlag

- Not robust
  - (or: tolerates 0 cheaters for correctness)

- Requires every server to participate
  - (and in the “right” order!)
Re-encryption Mixnet

0. Setup: mix servers generate a shared key

1. Users encrypt their inputs:

2. Encrypted inputs are mixed:

3. A quorum of mix servers decrypts the outputs

Recall: Discrete Logarithm Assumption

- \( p, q \) primes such that \( q \mid p-1 \)
- \( g \) is an element of order \( q \) and generates a group \( G_q \) of order \( q \)
- \( x \in \mathbb{Z}_q, y = g^x \mod p \)
- Given \((p, q, g, y)\), it is computationally hard to compute \( x \)
  - No polynomial time algorithm known
  - \( p \) should be 1024-bits and \( q \) be 160-bits
- \( x \) becomes the private key and \( y \) becomes the public key
ElGamal Encryption

- Encryption (of m in G_q):
  - Choose random r in Z_q
  - k = g^r mod p
  - c = my^r mod p
  - Output (k,c)

- Decryption of (k,c)
  - M = c * k^{-x} mod p

- Secure under discrete logarithm assumption

ElGamal Example: dummy

- Let’s construct an example

  KeyGen:
  - p = 11, q = 2 or 5; let’s say q = 5
  - 2 is a generator of Z_{11}^*
  - g = 2^2 = 4
  - x = 2; y = 4^2 mod 11 = 5

- Enc(3):
  - r = 4 \Rightarrow k = 4^4 mod 11 = 3
  - c = 3*5^4 mod 11 = 5

- Dec(3,5):
  - m = 5*3^{-2} mod 11 = 3
(t+1, n)- Secret Sharing

- Motivation: to secure the cryptosystem against $t \leq n/2$ corruptions
- Tool: Secret Sharing (Shamir’s Polynomial Secret Sharing)
  - any set of $t+1$ or more entities can recover the secret
  - an adversary who corrupts at most $t$ entities, learns nothing about the secret

**Tool: Shamir’s Polynomial Secret Sharing**

- $f(z) \rightarrow$ degree $t$ polynomial (mod $q$)
- $f(0) \rightarrow x$
- $f(i) \rightarrow ss[i]$

**INSECURE**

**SECURE**

Polynomial interpolation:

For any $G$, s.t. $|G| = t+1$

\[ x = \sum_{i \in G} ss[i] (\mod q) \]

(n=7, t=3)

---

Re-encryption technique

Input: a ciphertext $(k, c)$ wrt public key $y$

1. Pick a number $r'$ randomly from $[0...q-1]$
2. Compute
   \[ k' = kg^{r'} \mod p \]
   \[ c' = cy^{r'} \mod p \]
3. Output $(k', c')$

Same decryption technique!

Compute $m \leftarrow k' c^{t-x}$
A simple Mix

\[(k_1, c_1) \rightarrow (k'_1, c'_1) \rightarrow (k''_1, c''_1)\]
\[(k_2, c_2) \rightarrow (k'_2, c'_2) \rightarrow (k''_2, c''_2)\]
\[\ldots\]
\[(k_n, c_n) \rightarrow (k'_n, c'_n) \rightarrow (k''_n, c''_n)\]

Note: different cipher text, different re-encryption exponents!

And to get privacy... permute, too!

\[(k_1, c_1) \rightarrow \{\} \rightarrow \{\} \rightarrow (k''_1, c''_1)\]
\[(k_2, c_2) \rightarrow \{\} \rightarrow \{\} \rightarrow (k''_2, c''_2)\]
\[\ldots\]
\[(k_n, c_n) \rightarrow \{\} \rightarrow \{\} \rightarrow (k''_n, c''_n)\]
And, finally...the Proof

- Mix servers must prove correct re-encryption
  - Given \( n \) El Gamal ciphertexts \( E(m_i) \) as input
  - and \( n \) El Gamal ciphertexts \( E(m'_i) \) as output
  - Compute: \( E(\Pi m_i) \) and \( E(\Pi=m'_i) \)
  - Ask Mix for Zero-Knowledge proof that these ciphertexts decrypt to same plaintexts

Anonymizing Network in practice: Tor

- A low-latency anonymizing network
  http://www.torproject.org/
- Currently 1000 or so routers distributed all over the internet
- Peer-based: a client can choose to be a router
- A request is routed to/fro a series of a circuit of three routers
- A new circuit is chosen every 10 minutes
- *No real-world implementation of re-encryption mix as yet*