Scavenging for Anonymity with BlogDrop

Henry Corrigan-Gibbs       Bryan Ford
Yale University

Provable Privacy Workshop
9-10 July 2012 – Vigo, Spain
Motivation

• Alice is a citizen of country X
• Alice uses Tor to make an anonymous blog post to a server inside of country X
• Government of country X wants to find out post author’s identity

...how hard is that?
Motivation

• Tor average daily users in Q1 2012:
  ~49 000 in Iran
  ~16 000 in Syria
  ~2 000 in China

• Gov’t X can’t arrest thousands of people on a hunch

...what if the blog post has a timestamp?

Tor stats from https://metrics.torproject.org/
Internet Usage in a Day

If Alice is hiding among 3% of daily Tor users in China, she might be in trouble

AOL Web Search Data Set
Data mirrored at http://www.gregsadetsky.com/aol-data/

9 July 2012
Provable Privacy Workshop
State of the art

User Sessions

Anonymity set as large as the number of online users

9 July 2012 Provable Privacy Workshop
Outline

• Motivation
• **Overview: Anonymity scavenging**
• Ciphertext construction
• Conclusion
Anonymity Scavenging

- Can Alice increase latency to gain anonymity?
- High-latency systems are unpopular → unsafe
  - Mixmaster/mixminion vs. Tor
  - Would like low-latency Bobs to protect high-security Alices
  - Same motivation as alpha mixing (Dingledine et al. PETS’06)
Anonymity over time

User Sessions

time

9 July 2012

Provable Privacy Workshop
Anonymity over time

User Sessions

Blog A

9 July 2012
Anonymity over time

User Sessions

Blog B

9 July 2012
Provable Privacy Workshop
BlogDrop

Features
• Anonymous comm protocol in which user defines anonymity set size (vs. latency)
• High-security Alices hide amongst low-latency Bobs
• Accountable: protocol violations detectable

Assumptions
• At least one server is honest
• All users have pseudonym PK of blog author... more on this later
Bob’s Ciphertexts for Blogs A and B

Blog A

Blog B

Server X

Server Y

Server Z

9 July 2012

Provable Privacy Workshop
When each server has collected enough ciphertexts to satisfy **closure condition**, the servers each add their own ciphertext to the set.
Closure Condition

- How long do servers wait before revealing the plaintext message?
- Blog author picks a “closure condition”
  - After 9 July 2012 AND when there are 10 ciphertexts
  - After Alice, Bob, Carol, and Dave (identified by PKs) have all submitted ciphertexts
  - When there are $1\,000\,000$ in Swiss bank acct #098424713
  - Others...

→ Closure condition defines anon set
→ Poorly chosen closure conditions create anonymity risks... area for future work
Review

• Scavenging: Blog A and Blog B have different latencies and different anonymity set sizes
• One honest server enforces closure condition
• I omitted many details
  – e.g., Servers can flatten ciphertexts into an $O(L)$ size ciphertext — avoids $O(NL)$ storage
  – How servers agree on ciphertexts
  – ...
Outline

• Motivation
• Overview: Anonymity scavenging
• Ciphertext construction
• Conclusion
Ciphertext Construction

- Alice \((g^a)\)
- Server X \((g^x)\)
- Server Y \((g^y)\)
- Server Z \((g^z)\)

\[g^{ax} + g^{ay} + g^{az} + m = mg^{a(x+y+z)}\]

Using some group \(G = \langle g \rangle\) in which ElGamal cryptosystem is secure.

Client/server secret graph
(Chaum ’88) (Wolinsky et al., Eurosec’12)
Ciphertext Construction

Server X (g^x)
- g^{ax}

Server Y
- g^{ay}

Server Z
- g^{az}

+ 

m

mg^{a(x+y+z)}

Bob
- g^{bx}

- g^{by}

- g^{bz}

- g^{b(x+y+z)}

July 2012

Provable Privacy Workshop
Ciphertext Construction

Server X ($g^x$)
- Alice ($g^a$): $g^{ax}$
- Bob: $g^{bx}$
- Carol: $g^{cx}$

Server Y
- Alice ($g^a$): $g^{ay}$
- Bob: $g^{by}$
- Carol: $g^{cy}$

Server Z
- Alice ($g^a$): $g^{az}$
- Bob: $g^{bz}$
- Carol: $g^{cz}$

$m$

$mg^{a(x+y+z)}$

$g^{b(x+y+z)}$

$g^{c(x+y+z)}$
Ciphertext Construction

Alice \((g^a)\)

- Server X \((g^x)\): \(g^{ax}\)
- Server Y: \(g^{ay}\)
- Server Z: \(g^{az}\)

Bob

- Server X: \(g^{bx}\)
- Server Y: \(g^{by}\)
- Server Z: \(g^{bz}\)

Carol

- Server X: \(g^{cx}\) \(\rightarrow g^{-x(a+b+c)}\)
- Server Y: \(g^{cy}\) \(\rightarrow g^{-y(a+b+c)}\)
- Server Z: \(g^{cz}\) \(\rightarrow g^{-z(a+b+c)}\)

\[m\]

- \(mg^{a(x+y+z)}\)
- \(g^{b(x+y+z)}\)
- \(g^{c(x+y+z)}\)

Client/server secret graph

(Chaum’88) (Wolinsky et al., Eurosec’12)
Ciphertexts use iterative ElGamal encryption. Non-author plaintext=1

We exploit ElGamal’s multiplicative homomorphism to recover the plaintext

\[ \begin{align*}
    m & \\
    g^{-x(a+b+c)} & \\
    g^{-y(a+b+c)} & \\
    g^{-z(a+b+c)} & \\
    mg^{a(x+y+z)} & \\
    g^{b(x+y+z)} & \\
    g^{c(x+y+z)} & 
\end{align*} \]
Preventing Denial of Service

Assume that all users know anon author’s PK

PoK{ a, k: (C_{alice} = (g^xg^yg^z)^a \land A = g^a ) \lor K = g^k }

Alice knows the log of C_{alice} and that log is equal to her private key. i.e., Alice generated her ciphertext correctly

~ OR ~

Alice knows the author’s secret key and Alice can send whatever she wants

DoS-resistant DC-net (Golle and Juels, Eurocrypt’04)
Policy Document

• **The Catch 22:** To get anonymous communication, need to anonymously communicate the blog parameters
  – author’s pseudonym PK, closure condition, post length, etc

• Not quite: policy document only needs to be distributed once to set up blog

• e.g., Use once-per-month mix to shuffle policy documents
Outline

• Motivation
• Overview: Anonymity scavenging
• Ciphertext construction
• Conclusion
Conclusion

• Most existing systems allow user to be anonymous only among set of online users
• BlogDrop (via anonymity scavenging) gives anonymity among set of users over time
• High-security users hide amongst low-latency users
• DoS-resistant