DP5: PIR for Privacy-preserving Presence

Ian Goldberg    George Danezis

joint work with Nikita Borisov

Cryptography, Security, and Privacy Research Lab
University of Waterloo

University College London

31C3, 29 December 2014
Private information retrieval
Private information retrieval
Private information retrieval

PIR query

[Diagram showing a user and a database with an arrow indicating a PIR query]
Private information retrieval

PIR query

Ian Goldberg and George Danezis

DP5: PIR for Privacy-preserving Presence

2 / 24
A simple PIR protocol

\[ D = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 1 & \ldots & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & \ldots & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & \ldots & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & \ldots & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 1 \\
\vdots & & & & & & & \ddots & \vdots \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 1
\end{bmatrix} \]

If \( e_i = [0 \ 0 \ 1 \ 0 \ \ldots \ 0] \), then \( e_i \cdot D = v_1 \cdot D + v_2 \cdot D + \cdots + v_\ell \cdot D = (v_1 + v_2 + \cdots + v_\ell) \cdot D \)
A simple PIR protocol

\[ D = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 1 & \ldots & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & \ldots & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & \ldots & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & \ldots & 0 \\
\vdots & & & & & & & & \vdots \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 1
\end{bmatrix} \]

- If \( e_i = [0 \ 0 \ 1 \ 0 \ \ldots \ 0] \), then \( e_i \cdot D = \text{Block } i \)
- \( \mathbf{v}_1 \cdot D + \mathbf{v}_2 \cdot D + \cdots + \mathbf{v}_\ell \cdot D = (\mathbf{v}_1 + \mathbf{v}_2 + \cdots + \mathbf{v}_\ell) \cdot D \)
A simple PIR protocol

\[
D = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 1 & \ldots & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & \ldots & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & \ldots & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 1
\end{bmatrix}
\]

- If \( e_i = [0 \ 0 \ 1 \ 0 \ \ldots \ 0] \), then \( e_i \cdot D = \text{Block } i \)
- \( v_1 \cdot D + v_2 \cdot D + \cdots + v_\ell \cdot D = (v_1 + v_2 + \cdots + v_\ell) \cdot D \)
A simple PIR protocol

\[ D = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 1 & \ldots & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & \ldots & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & \ldots & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 1 & 1 & \ldots & 0 & 0 & 0 & \ldots & 1 \\
\end{bmatrix} \]

If \( e_i = [0 \ 0 \ 1 \ 0 \ \ldots \ 0] \), then
\[ e_i \cdot D = \text{Block}_i \]

\[ v_1 \cdot D + v_2 \cdot D + \ldots + v_\ell \cdot D = (v_1 + v_2 + \ldots + v_\ell) \cdot D \]

Previous work: variable-sized records
A simple PIR protocol

If $e_i = [0 \ 0 \ 1 \ 0 \ \ldots \ 0]$, then $e_i \cdot D = \text{Block } i$

$\mathbf{v}_1 \cdot D + \mathbf{v}_2 \cdot D + \cdots + \mathbf{v}_\ell \cdot D = (\mathbf{v}_1 + \mathbf{v}_2 + \cdots + \mathbf{v}_\ell) \cdot D$

Previous work: lookups by keyword or SQL
A simple PIR protocol

\[ D = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 1 & \ldots & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & \ldots & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & \ldots & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & \ldots & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & \ldots & 1
\end{bmatrix} \]

If \( e_i = [0 \ 0 \ 1 \ 0 \ \ldots \ 0] \), then \( e_i \cdot D = \text{Block } i \)

\( v_1 \cdot D + v_2 \cdot D + \cdots + v_\ell \cdot D = (v_1 + v_2 + \cdots + v_\ell) \cdot D \)

Previous work: robustness
Shamir secret sharing
Shamir secret sharing

$t = 1$

$t = 2$

$t = 3$
Shamir secret sharing

\[ t = 1 \]

\[ t = 2 \]

\[ t = 3 \]
Shamir secret sharing

$t = 1$

$t = 2$

$t = 3$
Error correction

$t = 3$
Error correction

\[ t = 3 \]
Error correction
Percy++ open-source library

git://git-crysp.uwaterloo.ca/percy

http://percy.sourceforge.net/
Social applications
Social applications
Social applications
Social applications

- Twitter
- Facebook
- XMPP
Online presence

![BuddyList example](image_url)

- Alfalfa: Away From My Desk
- Aretha
- Johnny: Online (Idle 90 Seconds)
- Winston: Working on Memoirs
- Darth: I Am Your Father
- EraserHead: On Vacation
- Felix
- Marvin: Thinking …
How it typically works

Authenticate

Bob, Charlie, ...

Bob, “Hi, Bob”

Alice, “Hi, Bob”

Ian Goldberg and George Danezis

DP5: PIR for Privacy-preserving Presence
How it typically works

Authenticate

⟨ Bob, Charlie, . . . ⟩

⟨ Bob, "Hi, Bob" ⟩

⟨ Alice, "Hi, Bob" ⟩
How it typically works

Authenticate

Bob, Charlie, ...

Bob, "Hi, Bob"

Alice, "Hi, Bob"
How it typically works
How it typically works

Authenticate

Bob, Charlie, …

Bob, “Hi, Bob”

Alice, “Hi, Bob”

Ian Goldberg and George Danezis

DP5: PIR for Privacy-preserving Presence
How it typically works

Authenticate

⟨Bob, Charlie, . . .⟩
⟨Bob, "Hi, Bob"⟩
⟨Alice, "Hi, Bob"⟩
How it typically works

Authenticate

Alice, "Hi, Bob"

Bob, Charlie, ...

XMPP
How it typically works

\[ \langle \text{Alice, "Hi, Bob"}, \ldots \rangle \]

\[ \langle \text{Bob, Charlie, \ldots} \rangle \]
How it typically works

authenticate

Bob, Charlie, ...
Bob, "Hi, Bob"

Alice, "Hi, Bob"

Ian Goldberg and George Danezis

DP5: PIR for Privacy-preserving Presence
How it typically works

\[\langle \text{Bob, "Hi, Bob"} \rangle\]
The problem
The problem
CONTROVERSIES

NSA Collects Online Address Books and Buddy Lists

The agency captures contacts when they’re transmitted across global servers, dodging domestic requirements mandating prior authorization for data collection inside the U.S.

By Courtney Subramanian @csub
Oct. 14, 2013
3 Comments

Senior intelligence officers and leaked documents from National Security Agency whistleblower Edward Snowden reveal that the NSA is amassing millions of contacts via online address books and instant-messaging buddy lists.

The program, under NSA's Special Source Operations branch, collects more than 250 million contacts in its database per year. A single day's data found that the agency accumulated 444,743 email address books from Yahoo, 105,068 from Hotmail, 82,857 from
“We kill people based on metadata”

General Michael Hayden, former Director of NSA

http://www.youtube.com/watch?v=UdQiz0Vavmc
Want: private presence

Presence features

Threat model

Security goals
Want: private presence

Presence features
- Friend registration
- Presence registration
- Presence status query
- Friend suspension / revocation

Threat model

Security goals
Want: private presence

Presence features
- Global passive adversary
- Dishonest users
- Secure end hosts
- Threshold of honest infrastructure servers
- Can’t break strong crypto

Threat model

Security goals
Want: private presence

Presence features
- Privacy of social network
- Privacy, integrity of presence and auxiliary data
- Unlinkability
- Suspension / revocation indistinguishable from offline
- Forward and backward secrecy
- Auditability

Threat model

Security goals
Why not ‘just use Tor’?
Why not ‘just use Tor’?

Register: A@anon.net

Tor

XMPP
Why not ‘just use Tor’?

Register: A@anon.net

Register: B@anon.net; Is A@anon.net on-line?

Tor

XMPP
Why not ‘just use Tor’?

Register: A@anon.net

Register: B@anon.net;
Is A@anon.net on-line?

Link between A@ and B@

Ian Goldberg and George Danezis
DP5: PIR for Privacy-preserving Presence
Why not ‘just use Tor’?

Register: A@anon.net

Register: B@anon.net;
Is A@anon.net on-line?

Link between A@ and B@

Ian Goldberg and George Danezis
Why not ‘just use Tor’?

‘Anonymous social graph’ isomorphic to real social graph → anonymization attacks.
- Easy to de-anonymize using side graphs (Remember Netflix!)

Pile-up the tricks?
- Do not register B@ – can still link all friends to a pseudonym.
- Use a separate circuit per since single friend? → Millions of circuits.
- ...

DP5 aims: do not require an anonymous channel; do not leak any social graph!
Introducing DP5 (High level idea)

DP5: PIR for Privacy-preserving Presence
Introducing DP5 (High level idea)

Private query!

Key idea: use private information retrieval for the lookup

Ian Goldberg and George Danezis
DP5: PIR for Privacy-preserving Presence
Introducing DP5 (High level idea)

Private query!

Key idea: use private information retrieval for the lookup

DP5

Aux. info.

Ian Goldberg and George Danezis

DP5: PIR for Privacy-preserving Presence
Introducing DP5 (High level idea)

Bob? Charlie?

DP5

Ian Goldberg and George Danezis
Introducing DP5 (High level idea)

⟨ Bob, Aux. info.⟩

DP5

Bob

Charlie

Private query!

Key idea: use private information retrieval for the lookup
Introducing DP5 (High level idea)

Private query!

Key idea: use private information retrieval for the lookup
Introducing DP5 (High level idea)

Bob? Charlie?

DP5

Private query!

Key idea: use private information retrieval for the lookup
Introducing DP5 (High level idea)

Private query!
Bob? Charlie?

Ian Goldberg and George Danezis
Key idea: use private information retrieval for the lookup

Private query!
DP5: Strawman version

\[ PRF_{Kab}(t_i) = PRF_{Kab}(t_i - 1) \cdot K_{\langle ID, C \rangle} \]

\[ AEAD(aux) \]

\[ PIR(C)^{-1} \cdot aux \]
\[ \text{PRF}^{K_{ab}}(t_i) \leftarrow \text{PRF}^{K_{ab}}(t_{i-1}) \]

\[ \langle \text{ID}, C \rangle \rightarrow \text{AEAD}(\text{aux}) \]
PRF_{K_{ab}}(t_i)
DP5: Strawman version

\[ PRF_{K_{ab}}(t_i) \xrightarrow{K} ID \]

\[ PRF_{K_{ab}}(t_i - 1) \xrightarrow{K} ID \]
DP5: Strawman version

\[ \text{PRF}_{K_{ab}}(t_i) \rightarrow \text{ID} \]

\[ \text{AEAD}_{K}^{(\text{aux})} \]
DP5: Strawman version

\[
\begin{align*}
\text{PRF}_{K_{ab}}(t_i) & \quad \rightarrow \quad \langle \text{ID}, C \rangle \\
\text{AEAD}_K^{(\text{aux})} & \quad \rightarrow \quad \text{PRF}_{K_{ab}}(t_i) \\
\end{align*}
\]
DP5: Strawman version

\[ \text{PRF}_{K_{ab}}(t_i) \]

\[ \langle \text{ID}, C \rangle \]

\[ \text{AEAD}_{K}^{(aux)} \]
DP5: Strawman version

$t_1$ $t_2$ $t_3$ $t_4$ …

$\text{PRF}^{K_{ab}}(t_i) \text{PRF}^{K_{ab}}(t_i - 1) K_{\langle ID, C \rangle}}$ 

$\text{AEAD}(\text{aux})$ (PIR) $(C)$

Ian Goldberg and George Danezis
DP5: Strawman version

\[ \text{PRF}_{K_{ab}}(t_{i-1}) \]
DP5: Strawman version

\[ PRF_{K_{ab}}(t_{i-1}) \leftarrow K \]

\[ PRF_{K_{ab}}(t_{i-1}) \leftarrow ID \]
DP5: Strawman version

\[ \text{PRF}_{K_{ab}}(t_i - 1) \rightarrow \text{ID} \]

(PIR)
DP5: Strawman version

\[ \text{PRF}_{K_{ab}}(t_{i-1}) \]

\[ \text{AEAD}(\text{aux}) \]

\[ \text{ID} \rightarrow \text{(PIR)} \]
DP5: Strawman version

\[ \text{PRF}_{K_{ab}}(t_{i-1}) \rightarrow \text{ID} \quad \text{(PIR)} \]

\[ \text{AEAD}_{K}^{-1}(C) \]

\[ \text{Ian Goldberg and George Danezis} \]

DP5: PIR for Privacy-preserving Presence 18 / 24
DP5: Strawman version

\[ \text{PRF}_{K_{ab}(t_{i-1})} \] 

\[ \text{PRF}_{K_{ab}(t_i)}(C) \] 

\[ \text{AEAD}^{-1}_K(C) \] 

\[ \text{AEAD}^{-1}_K(C) \] 

\[ \text{ID} \] 

(PIR)
The problem of the large database
The problem of the large database

David Wheeler
The problem of the large database

Any problem in computer science can be solved with another layer of indirection.

David Wheeler
Any problem in computer science can be solved with another layer of indirection.

But that will usually create another problem.
Two timescales, two databases
Two timescales, two databases

\[ T_1 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad \cdots \]

\[ T_2 \quad \cdots \]

\[ \langle \text{ID}, C \rangle \text{ AEAD} \quad \langle \text{ID}, C \rangle \text{ AEAD} \]

Ian Goldberg and George Danezis  
DP5: PIR for Privacy-preserving Presence  
20 / 24
Two timescales, two databases

\[ \begin{align*}
& T_1 \quad t_1 \quad T_2 \quad \ldots \\
& t_2 \quad t_3 \quad t_4 \quad \ldots 
\end{align*} \]

\[ \text{AEAD}_{K(P)} \]

\[ \text{PRF}_{K_{ab}(T_j)} \]

\[ \langle \text{ID}, C \rangle \]

\[ L \]
Two timescales, two databases

\[ T_1 \quad \cdots \quad T_2 \quad \cdots \]

\[ t_1 \quad t_2 \quad t_3 \quad t_4 \quad \cdots \]

\[ \text{PRF}_{K_{ab}}(T_{j-1}) \]

\[ \text{AEAD}^{-1}_K(C) \]

\[ \text{ID} \]

\[ \text{(PIR)} \]

\[ P \]

\[ L \]
Two timescales, two databases

\[ \begin{align*}
T_1 & \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad \cdots \\
& \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad \cdots \\
\end{align*} \]

\[ \text{PRF}_{P(t_i)} \rightarrow \langle \text{ID}, C \rangle \]

\[ \text{AEAD}_K^{(aux)} \]

\[ S \]
Two timescales, two databases

\[
T_1 \quad T_2 \quad \cdots
\]

\[
t_1 \quad t_2 \quad t_3 \quad t_4 \quad \cdots
\]

\[
PRF_{ab}(T_j) \quad (PIR)(C)
\]

\[
\text{AEAD}_{-1}^k(C)
\]

\[
PRF_P(t_{i-1})
\]

\[
\text{ID} \quad \text{(PIR)}
\]

\[
S
\]

Ian Goldberg and George Danezis

DP5: PIR for Privacy-preserving Presence
Implementation

PIR: Percy++ PIR library (C++)

DP5 core: C++, Python Bindings

Networking: Cherrypy framework, Twisted (Python)

Missing: Integration into common chat clients
Cost of running a DP5 PIR server

(Long-term database, 24-hour epoch)

Number of users

Monthly per-user cost ($)

Bandwidth

CPU

Total
Takeaways

- Metadata in social communication is being targeted

- Private information retrieval (PIR) allows database lookups without revealing the query to the database servers themselves

- DP5 uses PIR to achieve private presence—people learn when their friends are online (and how to contact them securely) without any server ever learning who is friends with whom
Find out more

- Technical report

- Git code repository
  git://git-crysp.uwaterloo.ca/dp5