:: Privacy Pass ::

Bypassing internet challenges anonymously

Alex Davidson¹,³  Ian Goldberg²  Nick Sullivan³
George Tankersley⁴  Filippo Valsorda⁴

¹Royal Holloway, University of London ²University of Waterloo ³Cloudflare ⁴Independent

PETS 2018, Barcelona
July 25, 2019

https://privacypass.github.io

alex.davidson.2014@rhul.ac.uk // @alxdavids
Background

Anonymous authentication protocol

Privacy Pass

Summary
Content delivery networks

e.g. DDoS, spam filtering, content scraping etc...
Content delivery networks

e.g. DDoS, spam filtering, content scraping etc...
Content delivery networks

e.g. DDoS, spam filtering, content scraping etc...
Content delivery networks

e.g. DDoS, spam filtering, content scraping etc...
IP reputation

User
←
W
→
CDN

27.2.187.41
27.2.187.41
IP reputation

User

⊥

CDN

27.2.187.41
Is this a good system?

::false negatives::

particularly users of static, shared IP addresses
Is this a good system?

::affected users::
Is this a good system?

:::worst case:::

27.2.187.41
Is this a good system?

::average case::

User → CDN

27.2.187.41
Is this a good system?

::average case::

User \rightarrow CDN

27.2.187.41
Is this a good system?

::average case::

User ➔ W ➔ CDN

27.2.187.41
Problems with challenges (aka CAPTCHAs)

::: Heavily JS reliant

::: Potentially block access

::: Annoying/hard

::: Slow

::: Questionable protection

::: More round trips
Possible solutions

::no blocking::

User \[\rightsquigarrow\] CDN
Possible solutions

::cookies?::

User ← W → CDN
Possible solutions

::cookies?::

problem: linkability
Contributions

::: Anonymous authentication protocol
    ::: based on elliptic curves and oblivious prfs
    ::: combination of prior techniques [JKK14, Hen14]

::: Client-side implementation in browser extension

::: Server-side deployment in Cloudflare edge servers

::: Empirical survey of results
Background

Anonymous authentication protocol

Privacy Pass

Summary
Oblivious pseudorandom function (OPRF)
Oblivious pseudorandom function (OPRF)

$x$ is hidden from the PRF evaluator
Oblivious pseudorandom function (OPRF)

$K$ is not revealed to $C$
Verifiable OPRF (VOPRF)

\[ \pi \text{ is a NIZK proof that } y \leftarrow \text{PRF}(K,x) \]
Elliptic curve VOPRF (EC-VOPRF)

\[ [x] = H(x)^r \]

\[ Y = [x]^k \]

\[ \pi = \text{DLEQ} \]

H hashes x to an elliptic curve

\[ \pi \text{ is a discrete log equivalence (DLEQ) proof} \]
DLEQ proofs

::summary::

public commitments: $g, h = g^k$

signed token pair: $x, y$

show that $\log_g(h) = \log_x(y) = k$

without revealing $k$
Anonymous authentication protocol

::signing::

$[x] \rightarrow \text{Server}$
Anonymous authentication protocol

::signing::

C

\[ H(x)^k \]

Y

\[ \pi \]

Server
Anonymous authentication protocol

::redemption::

server verifies MAC to authenticate C
Anonymous authentication protocol

::multiple tokens::

C \rightarrow \text{Server}

\{[x_i]\}_i \rightarrow \{\pi_i\}_i \rightarrow \{y_i\}_i
Anonymous authentication protocol

::multiple tokens::

\[ C \rightarrow \text{Server} \]

\[ \{y_i\}_i \quad \{\pi_i\}_i \]

similar design to [JKK14]
Anonymous authentication protocol

:: multiple tokens ::

\[
\begin{align*}
\text{C} & \quad \{y_i\}_i \quad \pi \\
\text{Server} &
\end{align*}
\]

batched DLEQ proofs! [Hen14]
Security properties

::unlinkability::

::: any x should be unlinkable from any signing phase

::: prevents server from linking authentication sessions

::: \( H(x)^r \) uniformly blinds x from Server
Security properties

::: one-more-token security :::

::: for \( N \) signed tokens, hard to create \( N + 1 \) signed tokens

::: prevents client from forging signed tokens

::: reduction from one-more-decryption security of El Gamal
Security properties

::: Key consistency:::

::: ensures that all tokens are signed by one key k
::: prevent server deanonymisation using different keys
::: soundness of batch DLEQ proof [Hen14]
Background

Anonymous authentication protocol

Privacy Pass

Summary
Privacy Pass

::browser extension::
Privacy Pass

::: Cloudflare:::

::: CDN serves 10% of internet traffic
::: use CAPTCHAs to prevent bots accessing origins
::: use IP reputation to decide challenging or not
Privacy Pass

::acquiring signed tokens::

\{x_i\}_i \rightarrow k
Privacy Pass

::acquiring signed tokens::

\{x_i\}_i \rightarrow \{[x_i]\}_i \rightarrow k
Privacy Pass

::acquiring signed tokens::

\[ \bar{W}, \{y_i\}_i, \pi \]

\[ \{x_i\}_i \quad \{H(x_i)^k\}_i \quad k \]
Privacy Pass

::bypassing challenges::

\[ \{x_i\}_i \quad \{H(x_i)^k\}_i \]

\[ k \]
Privacy Pass

::bypassing challenges::

\[ \{x_i\}_i \rightarrow \{H(x_i)^k\}_i \rightarrow MAC_i \rightarrow k \rightarrow x_i \]
Privacy Pass

::bypassing challenges::

\[
\{x_i\}_i \quad \text{\{H(x_i)^k\}_i} \quad \hat{W} \quad \text{SDC} \quad k \quad x_j
\]
Specifics

::: Elliptic curve: NIST P256

::: Public commitments \((g, g^k)\) for DLEQ verification

::: Batch DLEQ PRNG: SHAKE-256

::: Default # of signed tokens (client-side): 30

::: Max signed tokens (server-side): 300

::: Triggers: \{status codes, headers\}

::: Code:

::: https://github.com/privacypass/challenge-bypass-extension
::: https://github.com/privacypass/challenge-bypass-server
::: https://privacypass.github.io/protocol (protocol summary)
## Benchmarks

::Timings (ms)::

<table>
<thead>
<tr>
<th>Operation</th>
<th>Timings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token generation</td>
<td>$120 + 64 \cdot N$</td>
</tr>
<tr>
<td>Verify DLEQ</td>
<td>$220 + 110 \cdot N$</td>
</tr>
<tr>
<td><strong>Total signing request</strong></td>
<td>$340 + 180 \cdot N$</td>
</tr>
<tr>
<td><strong>Total redeem request</strong></td>
<td>57</td>
</tr>
<tr>
<td><strong>Client</strong></td>
<td></td>
</tr>
<tr>
<td>Signing</td>
<td>$0.04 + 0.20 \cdot N$</td>
</tr>
<tr>
<td>DLEQ generation</td>
<td>$0.32 + 0.55 \cdot N$</td>
</tr>
<tr>
<td><strong>Total signing</strong></td>
<td>$1.48 + 0.87 \cdot N$</td>
</tr>
<tr>
<td><strong>Total redemption</strong></td>
<td>0.8</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td></td>
</tr>
</tbody>
</table>

$N = \# \text{ of tokens batch signed}$
## Benchmarks

### Request size (bytes):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signing request (U → CDN)</td>
<td>$57 + 63 \cdot N$</td>
</tr>
<tr>
<td>Signing response (CDN → U)</td>
<td>$295 + 121 \cdot N$</td>
</tr>
<tr>
<td>Redemption request (U → CDN)</td>
<td>396</td>
</tr>
</tbody>
</table>

$N = \# \text{ of tokens batch signed}$
Cloudflare deployment (Nov 2017)

::Release::

::: Extension released: 8 Nov 2017

::: Downloads (28 Nov 2017)

:: Chrome extension: 8499

:: Firefox add-on: 3489

::: Downloads (Jul 2018)

:: Chrome extension: 61578

:: Firefox add-on: 16375
Cloudflare deployment (Nov 2017)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Global</th>
<th>Tor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total requests (per week)</td>
<td>1.6 trillion</td>
<td>700 million</td>
</tr>
<tr>
<td>Total challenged requests</td>
<td>1.04%</td>
<td>17%</td>
</tr>
<tr>
<td>Signs (peak per hour)</td>
<td>～600</td>
<td>～100</td>
</tr>
<tr>
<td>Redeems {Nov 2017} (peak per hour)</td>
<td>～2000</td>
<td>～200</td>
</tr>
<tr>
<td>Redeems {Jul 2018} (peak per hour)</td>
<td>～3300</td>
<td>～600</td>
</tr>
<tr>
<td>Single-domain cookies (Nov 2017)</td>
<td>515 million</td>
<td>34 million</td>
</tr>
</tbody>
</table>
Background

Anonymous authentication protocol

Privacy Pass

Summary
Conclusion and links

::: Privacy Pass extension is still in **beta**
::: Further analysis of protocol/code would be welcome!
Conclusion and links

::: Privacy Pass extension is still in **beta**
::: Further analysis of protocol/code would be welcome!
::: Protocol spec:
    ::: https://tinyurl.com/pp-protocol
::: Website:
    ::: https://privacypass.github.io
::: Code (contribute!):
    ::: https://github.com/privacypass/challenge-bypass-extension
    ::: https://github.com/privacypass/challenge-bypass-server
::: Support:
    ::: privacy-pass-support@cloudflare.com
Final notes

::: See paper for:
   { more analysis of out-of-band attacks, comparison with existing research, security proofs, implementation details }

::: EC-VOPRF IETF standardisation
    :: https://github.com/chris-wood/draft-sullivan-cfrg-voprf

::: Future work:
   { DLEQ update, more integrations, better documentation, PQ VOPRF }
Final notes

::: See paper for:
    { more analysis of out-of-band attacks, comparison with existing research, security proofs, implementation details }

::: EC-VOPRF IETF standardisation
    :: https://github.com/chris-wood/draft-sullivan-cfrg-voprf

::: Future work:
    { DLEQ update, more integrations, better documentation, PQ VOPRF }

Thanks for listening!

https://privacypass.github.io
References

*Efficient Zero-Knowledge Proofs and Applications.*
http://hdl.handle.net/10012/8621.

Round-optimal password-protected secret sharing and T-PAKE in the password-only model. 