:: Privacy Pass ::

Bypassing internet challenges anonymously

Alex Davidson [1,3] Ian Goldberg [2] Nick Sullivan [3]
George Tankersley [4] Filippo Valsorda [4]

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

PETS 2018, Barcelona
July 25, 2019

https://privacypass.github.io

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

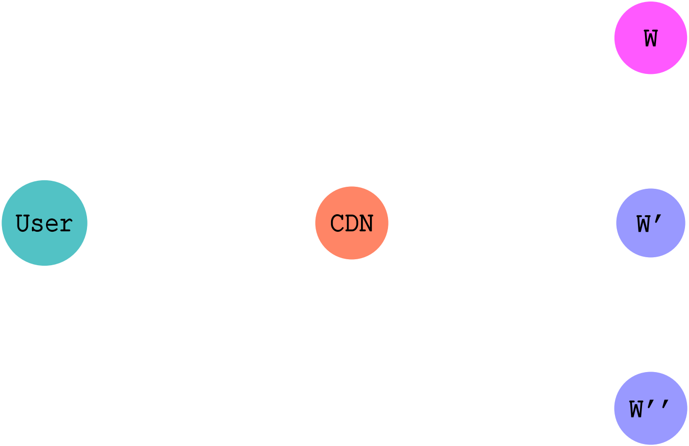
Background

Anonymous authentication protocol

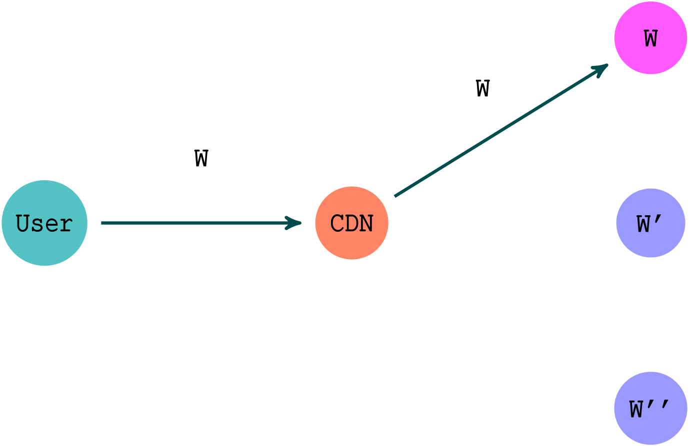
Privacy Pass

Summary

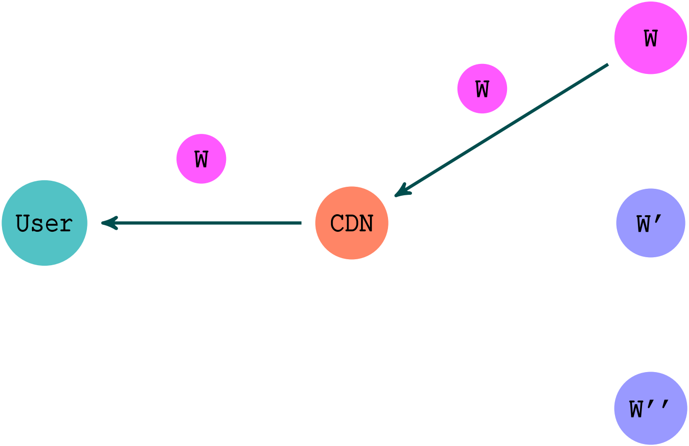
Content delivery networks



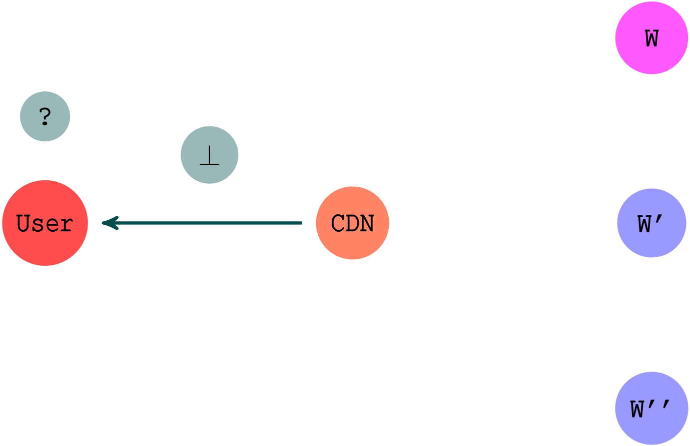
Content delivery networks



Content delivery networks

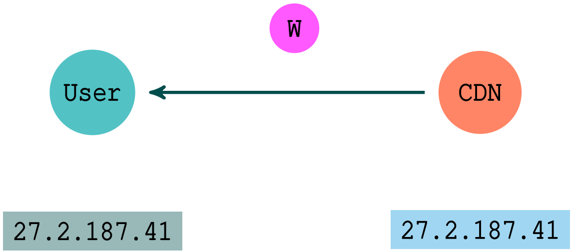


Content delivery networks

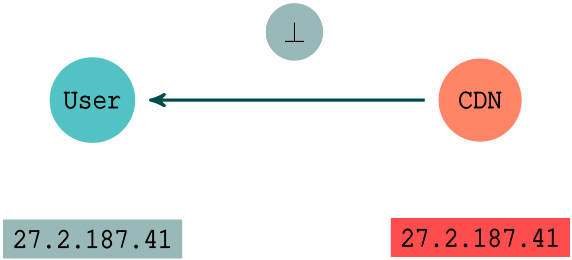


e.g. DDoS, spam filtering, content scraping etc...

IP reputation



IP reputation



Is this a good system?

::false negatives::

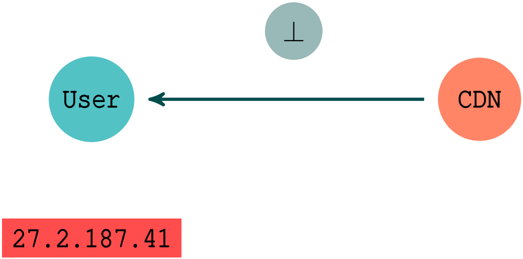
particularly users of static, shared IP addresses

Is this a good system?

::affected users::

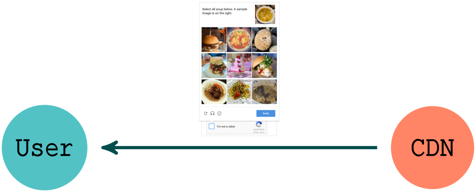
Is this a good system?

::worst case::



Is this a good system?

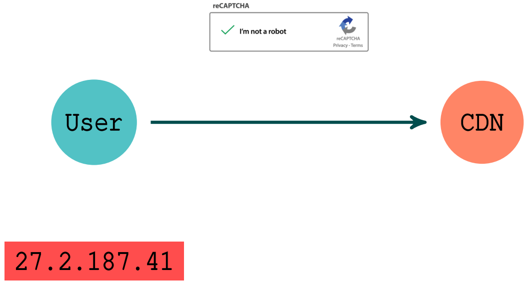
::average case::



27.2.187.41

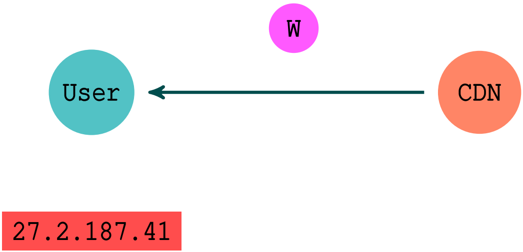
Is this a good system?

::average case::

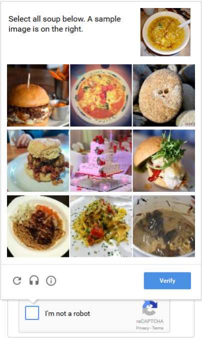


Is this a good system?

::average case::



Problems with challenges (aka CAPTCHAs)



::: Heavily JS reliant

::: Potentially block access

::: Annoying/hard

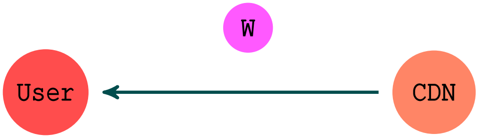
::: Slow

::: Questionable protection

::: More round trips

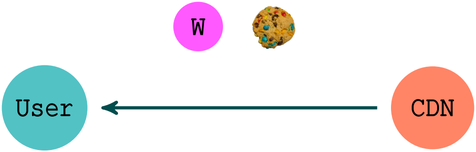
Possible solutions

::no blocking::



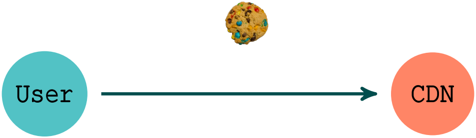
Possible solutions

::cookies?::



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

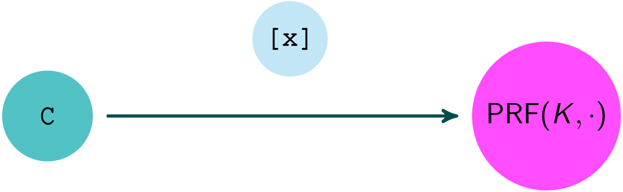
Anonymous authentication protocol

Privacy Pass

Summary

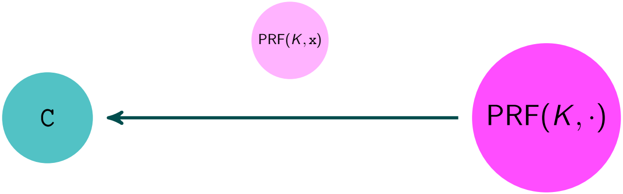
Oblivious pseudorandom function (OPRF)

Oblivious pseudorandom function (OPRF)



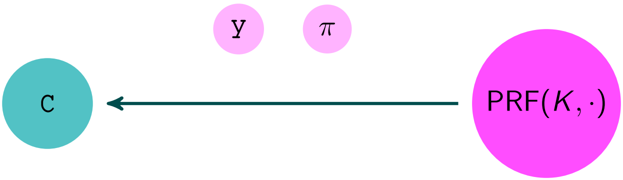
x is hidden from the PRF evaluator

Oblivious pseudorandom function (OPRF)



K is not revealed to C

Verifiable OPRF (VOPRF)



π is a NIZK proof that $\mathtt{y} \leftarrow \mathsf{PRF}(K,\mathtt{x})$

Elliptic curve VOPRF (EC-VOPRF)

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

$$y = [x]^k$$

$$\pi = \text{DLEQ}$$

H hashes x to an elliptic curve

π is a discrete log equivalence (DLEQ) proof

DLEQ proofs

::summary::

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

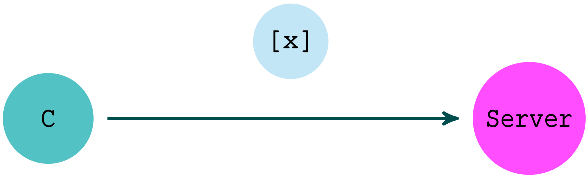
signed token pair: x, y

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

without revealing k

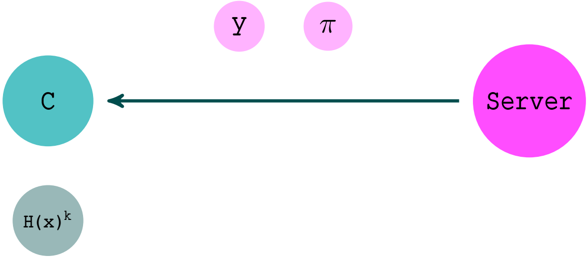
Anonymous authentication protocol

::signing::



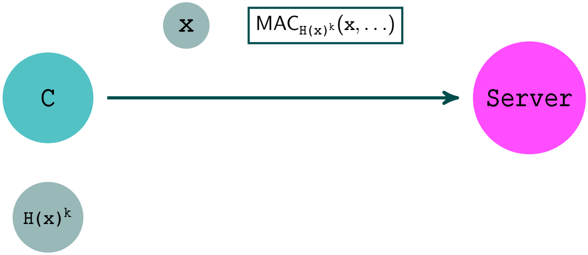
Anonymous authentication protocol

::signing::



Anonymous authentication protocol

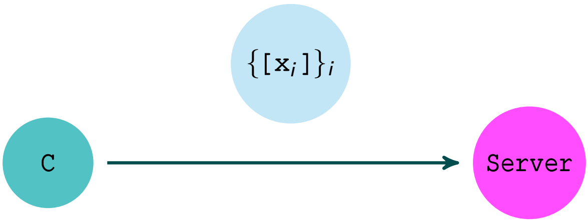
::redemption::



server verifies MAC to authenticate C

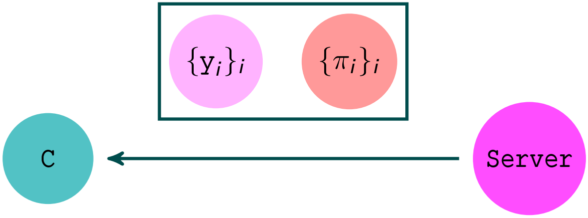
Anonymous authentication protocol

::multiple tokens::



Anonymous authentication protocol

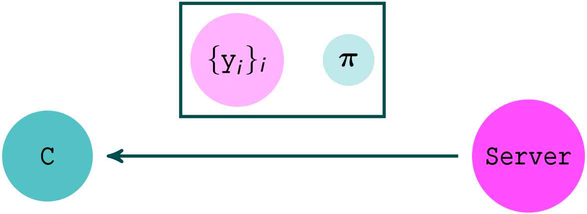
::multiple tokens::



similar design to [JKK14]

Anonymous authentication protocol

::multiple tokens::



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

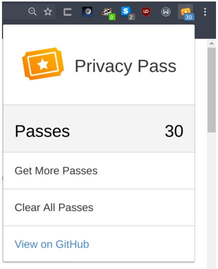
::Key consistency::

::: ensures that all tokens are signed by one key k

::: prevent server deanonymisation using different keys

::: soundness of batch DLEQ proof [Hen14]

Privacy Pass



::browser extension::

Privacy Pass

::Cloudflare::

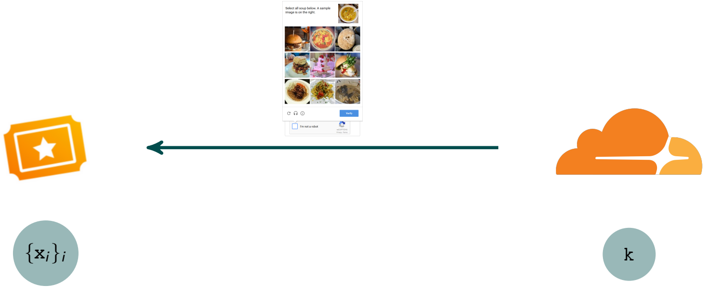
::: CDN serves 10% of internet traffic

::: use CAPTCHAs to prevent bots accessing origins

::: use IP reputation to decide challenging or not

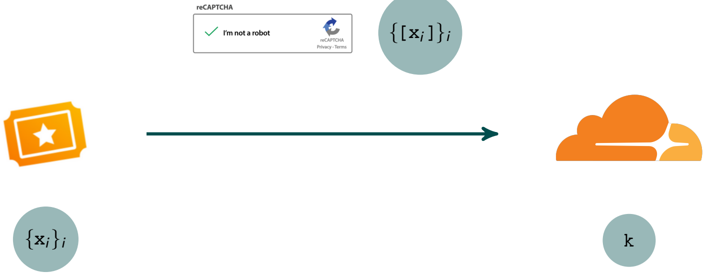
Privacy Pass

::acquiring signed tokens::



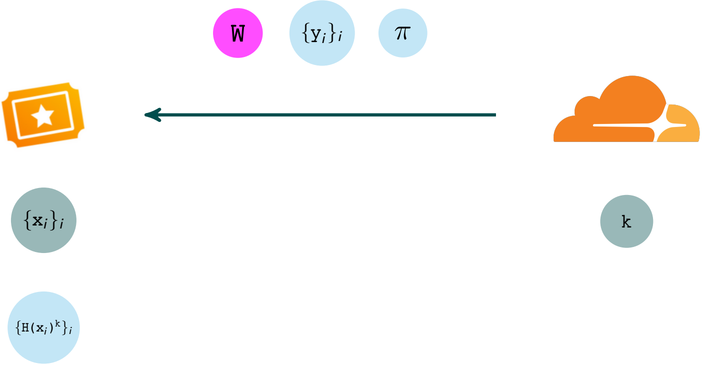
Privacy Pass

::acquiring signed tokens::



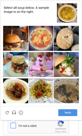
Privacy Pass

::acquiring signed tokens::



Privacy Pass

::bypassing challenges::



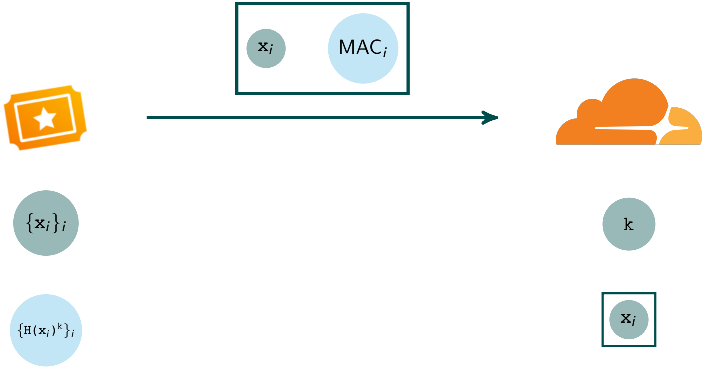
$\{x_i\}_i$

k

$\{H(x_i)^k\}_i$

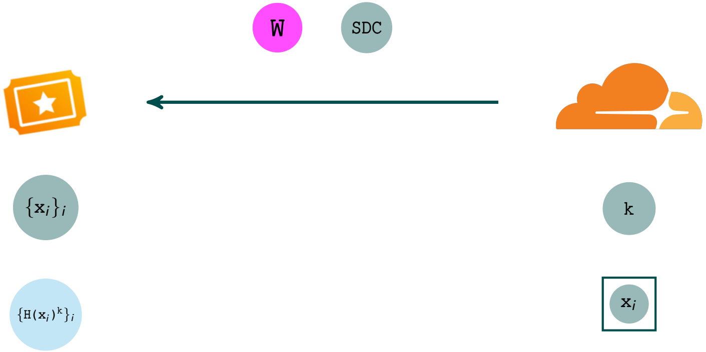
Privacy Pass

::bypassing challenges::



Privacy Pass

::bypassing challenges::



Specifics

::: Elliptic curve: NIST P256

::: $\boxed{\text{Public commitments } (g, g^k) \text{ for DLEQ verification}}$

::: Batch DLEQ PRNG: SHAKE-256

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

::: $\boxed{\text{Max signed tokens (server-side): 300}}$

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

::: 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)::

	Operation	Timings
Client	Token generation	$120 + 64 \cdot N$
	Verify DLEQ	$220 + 110 \cdot N$
	Total signing request	$340 + 180 \cdot N$
	Total redeem request	57
Server	Signing	$0.04 + 0.20 \cdot N$
	DLEQ generation	$0.32 + 0.55 \cdot N$
	Total signing	$1.48 + 0.87 \cdot N$
	Total redemption	0.8

$N =$ # of tokens batch signed

Benchmarks

::Request size (bytes)::

Operation	Size (bytes)
Signing request (U $\rightarrow$ CDN)	$57 + 63 \cdot N$
Signing response (CDN $\rightarrow$ U)	$295 + 121 \cdot N$
Redemption request (U $\rightarrow$ CDN)	396

$N =$ # 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)

Metric	Global	Tor
Total requests (per week)	1.6 trillion	700 million
Total challenged requests	1.04%	17%
Signs (peak per hour)	$\sim$600	$\sim$100
Redeems {Nov 2017} (peak per hour)	$\sim$2000	$\sim$200
Redeems {Jul 2018} (peak per hour)	$\sim$3300	$\sim$600
Single-domain cookies (Nov 2017)	515 million	34 million

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

[Hen14] Henry, Ryan.
Efficient Zero-Knowledge Proofs and Applications.
PhD thesis, University of Waterloo, 2014.
http://hdl.handle.net/10012/8621.

[JKK14] Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk.
Round-optimal password-protected secret sharing and T-PAKE in the password-only model.
In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 233--253. Springer, Heidelberg, December 2014.