Anonymous rate limiting with Direct Anonymous Attestation

Alex Catarineu
Philipp Claßen
Konark Modi
Josep M. Pujol

Cliqz GmbH, Munich

Data is essential to build services

Problems with Data Collection

IP	UA	Timestamp	Message Type	Payload	Cookie
195.202.XX.XX	FF..	2018-07-09 14:01	QueryLog	[face, facebook.com]	Cookie=966347bfd1e550
195.202.XX.XX	Chrome..	2018-07-09 14:06	Page	https://analytics.twitter.com/user/konarkmodi	Cookie=966347bfd1e55040434abe…
195.202.XX.XX	Chrome..	2018-07-09 14:10	QueryLog	[face, facebook.com]	Cookie=966347bfd1e55040434abe…
195.202.XX.XX	Chrome..	2018-07-09 16:15	Page	https://booking.com/hotels/barcelona	Cookie=966347bfd1e55040434abe…
195.202.XX.XX	Chrome..	2018-07-09 14:10	QueryLog	[face, facebook.com]	Cookie=966347bfd1e55040434abe…
195.202.XX.XX	FF..	2018-07-09 18:40	Page	https://shop.flixbus.de/user/resetting/reset/**hi7KTb1Pxa4lXqKMcwLXC0XzN-47Tt0Q**	Cookie=966347bfd1e55040434abe…

Anonymous data collection

Timestamp	Message Type	Payload
2018-07-09 14	Querylog	[face, facebook.com]
2018-07-09 14	Querylog	[boo, booking.com]
2018-07-09 14	Page	https://booking.com/hotels/barcelona
2018-07-09 14	Telemetry	[‘engagement’: 0 page loads last week, 5023 page loads last month]

More details:
https://gist.github.com/solso/423a1104a9e3c1e3b8d7c9ca14e885e5
http://josepmpujol.net/public/papers/big_green_tracker.pdf

Motivation: Preventing attacks on anonymous data collection

Timestamp	Message Type	Payload
2018-07-09 14	querylog	[book, booking.com]
2018-07-09 14	querylog	[fac, facebook.com]
…	….	…
…	….	…
2018-07-09 16	querylog	[ama, amazon.de]
2018-07-09 18	querylog	[book, booking.com]
2018-07-09 18	querylog	[bookin, booking.com]

Genuine or Bogus ?

< booking, https://www.booking.com/, ts: 2018-07-19 14 >

< booking, https://www.bookingholidays.com/, ts: 2018-07-19 14 >

Anonymous rate-limiting service (V1)

- In production since 2016 for over 2mill. DAU
- Tightly coupled to Cliqz use cases.
- Proxies performed an extra task of rate-limiting
  - Needed to run custom application.
  - Needed multiple trusted 3rd parties to prevent from colluding with other proxies and Cliqz
- Could not use off the shelf commercial solutions or Tor for network anonymity
- Deprecated in favour of V2

Anonymous rate-limiting (v2)

- Based on Direct Anonymous Attestation.
- Benefits:
  1. No need for trusted third parties.
  2. Less interactions between client and server.
  3. Can use known solutions (Tor...) for network anonymity.

Direct Anonymous Attestation (DAA)

- Implemented in TPM standard (and others via EPID spec).

Key features:

1. Anonymous authentication: prove that a TPM is signing a message, not **which** one.
2. Controlled linkability: two signatures from same member/TPM are linkable iff they are done w.r.t. same **basename** string.

- We can use the linkability properties to achieve rate-limiting.

DAA operations

- **Join:** A device gets credentials from **Issuer** (becomes a member of the Issuer group).
- **Sign:** A **Member** can sign messages w.r.t. a *basename* string.
- **Verify:** A **Verifier** can check whether a signature is valid for a given message, basename
- **Link:** A Verifier can check whether two signatures have been done by the same member and the same basename.

DAA Signature Linkability

- Signatures contain a **pseudonym** = "linkability tag"
- pseudonym = OneWay(basename, userPrivKey)
- Equal pseudonyms mean same basename and member.
- Link algorithm is efficient.

Rate-limiting with DAA

- Define a "language" for the basenames to enforce concrete rate-limiting rules.
- Client computes basenames according to the rules and never sends two messages with the same basename/pseudonym.
- Verifier drops messages with incorrect basename or repeated pseudonym.

Rate-limiting with DAA

Basename: <message_type, time_period, message_digest, count>

Time period: a truncated timestamp (hourly, daily, etc.)
Message digest: a function of the message (e.g. search query)
Count: 1 <= count <= N, a multiplier (can send N messages for a given prefix)

Expressive enough for our needs, but other structures are possible

Example 1: Query Log

Original Message

< search_query: BOokinG, landing_url: http://www.booking.com, ts: 2018-07-19 18:56 >

Limit rule

one message per **user**, per **hour**, per *normalized* search **query**

Basename

<querylog_type, 2018-07-19 18:00, booking, 1>

Example 2: three daily messages

- 3 unique basenames (independent of message content):

<threedaily_type, 2018-07-19, "", 3>
<threedaily_type, 2018-07-19, "", 1>
<threedaily_type, 2018-07-19, "", 2>

- Counts should be chosen at random from the unused ones.

Problem: we do not always have a TPM

- Our use case: client/member runs in a web environment (browser extension).
- Other usages for rate-limiting may also have limited execution environments.
- Not realistic to assume a TPM is available.

*At least for now.

Let's remove the TPM

- Implement DAA without TPM, software only.
- Problem: an attacker can create many identities and obtain credentials for all of them.
- If we cannot limit number of credentials given to attackers, whole system will be useless.

Sybil attacks

- Rate-limit the Join operation.
- Note: Joins do not need to be anonymized.
- Can leverage standard techniques, just on Join (e.g. rate-limiting via IP).
- Depending on use case, email, user accounts might be used.
- Extra measure: rotate Issuer public keys.
- We believe there are ways to mitigate Sybils, and that the rate-limiting system is useful even without TPMs.

The three key parts of our system

- Rate-limit **Joins** using all available info
- Rotate Issuer keys periodically
- Use DAA linkability for message rate-limiting

Possible deanonymization attacks (1)

- Different Issuer/Group public keys for every user.
  - One group per user.
  - Quite naïve, easily detected: user fetches Issuer public keys anonymously

Possible deanonymization attacks (2)

- Denial of Join:
  - Issuer can allow to join only selected users.
  - Anonymity set reduced -> less users with valid credentials.
  - More difficult to detect, but possible.
  - Incentive to track users vs. Incentive to obtain data to run services.

Similar systems

- Privacy Pass (https://privacypass.github.io)
  - Used in Cloudflare.
  - Users receive anonymous tokens that can redeem to avoid solving more CAPTCHAs.
- Anonize (https://anonize.org)
  - Implemented in Brave Browser.
  - Anonymous surveys (users can vote only once per survey).
  - Used for Payments, to decide popularity of websites.

Implementation

- Based on [FIDO-DAA-Security-Proof]: pairing-based cryptography.
- Implemented in C using ***Apache Milagro Crypto Library*** [milagro-crypto-c].
- Our use case: user/client runs in a **web browser**, issuer/verifier/collector is a server controlled by us.
- Compiled to **WebAssembly** with bindings to JavaScript for the client.
- Native bindings for NodeJS server (faster than WebAssembly version).

Benchmarks

- Signature size (msg overhead): 379 bytes (3112 bits)
- Client join: ~50 ms
- Server process join: ~2 ms
- Client sign: ~7 ms
- Server verify: ~4 ms
  - 250 messages/second per CPU core

Thank you for listening

- https://github.com/cliqz-oss/anonymous-credentials
- https://github.com/cliqz-oss/browser-core