Counting with STAR
Shipping A Privacy-Preserving Telemetry System To Millions Of Users

Shivan Kaul Sahib
@shivan_kaul
Brave Software

USENIX PEPR ‘23
What is STAR?

1. Secret Sharing for Private Threshold Aggregation Reporting
[CCS ’22]
What is STAR?

1. Secret Sharing for Private Threshold Aggregation Reporting
   [CCS ’22]
2. “Balls of gas burning billions of miles away”
STAR life-cycle

1. **Nebula**: why STAR
2. **Protostar**: designing STAR
3. **Red giant**: shipping & scaling
4. **Going supernova**: new features unlocked!
5. **Black holes**: and how to avoid them
"Don’t sell my data"
Sign in or create your account

Not sure if you have an account?
Enter your email and we’ll check for you.

Email Address

Continue

Securing your personal information is our priority.
See our privacy measures.
Give users a way to report https://walmart.com as broken
Give users a way to report https://walmart.com as broken with some amount of privacy cover
Generalizing...

Give clients a way to report arbitrary strings to the server in a privacy-preserving way.
Options

1. MPC-based (Prio, Poplar)
   a. Too slow and/or expensive
   b. Complicated to deploy
   c. Bad failure mode under collusion
Options

1. MPC-based (Prio, Poplar)
   a. Too slow and/or expensive
   b. Complicated to deploy
   c. Bad failure mode under collusion

Takeaway: simplicity is important for user trust
Options

1. MPC-based (Prio, Poplar)
2. Differential privacy-based (RAPPOR, IPA)
   a. (central) not user-auditable
   b. Hard to test
   c. (central) Bad failure mode
   d. (local) Requires large user base
   e. Hard to use for exact strings
Options

1. MPC-based (Prio, Poplar)
2. Differential privacy-based (RAPPOR, IPA)

**Takeaway:** once data leaves device, all bets are off
Generalizing...

Give clients a way to report arbitrary strings to the server only if N other users also reported that string.
STAR goals

Cheap
Simple
Private
STAR goals

Cheap
Simple
Private

Takeaway: clarity of your unique requirements can lead to generally useful systems!
Protostar.
1. K-threshold aggregation scheme
2. Secret sharing
3. OPRFs (oblivious pseudorandom functions) to derive randomness for low entropy input space
Client wants to send “walmart.com”

1. Hash “walmart.com” to get random value => R
2. Derive symmetric key $S$ from $R = \text{derive}(R)$
3. Encrypts message “walmart.com” using $S$: $M = \text{Encrypt}(S, \text{“walmart.com”})$
4. Generate secret share of $S$: $\text{SecretShareOfS_i}$
5. Client sends server: $\{M, \text{SecretShareOfS_i}\}$
User 1: M, SecretShareOfS1
User 2: M, SecretShareOfS2
User 3: M, SecretShareOfS3

K = 3
1. Recover encryption key: $S = \text{Recover}(\text{SecretShareOf}S_{i..K})$

2. Use $S$ to decrypt $M$:

   "walmart.com" = Decrypt($S, M$)
STAR goals

1. Cheap
   a. 24x cheaper than alternatives [1]

2. Simple
   a. Straightforward protocol
   b. One server model (when randomness derived locally)

3. Private
   a. K value is verifiable by clients
Red Giant.

shipping & scaling
Shipping & Scaling

1. Evolved the protocol
2. Many, many prototypes
3. Academia (PPORPF) + standards (verifiability)
4. First deployment was specifically for one use-case (JS)
5. Expanded out to more general C++ browser support
6. Went through security & privacy review
7. Open-sourced
8. Whole process took a year
Shipping & Scaling

1. Evolved the protocol
2. Many, many prototypes
3. Academia (PPORPF) + standards (verifiability)
4. First deployment was specifically for one use-case (JS)
5. Expanded out to more general C++ browser support
6. Went through security & privacy review
7. Open-sourced
8. Whole process took a year

Takeaway: boring crypto is good!
Shipping & Scaling

1. Evolved the protocol
2. Many, many prototypes
3. Academia (PPORPF) + standards (verifiability)
4. First deployment was specifically for one use-case (JS)
5. Expanded out to more general C++ browser support
6. Went through security & privacy review
7. Open-sourced
8. Whole process took a year

**Takeaway:** academic/standards process can improve your system
Supernova.

unlocking new features
New features

1. Web Discovery Project
2. Ref codes
New features

1. Web Discovery Project
2. Ref codes

Takeaway: telemetry systems are surprisingly useful for many things!
New features

1. Web Discovery Project
2. Ref codes
3. Country codes? No!
Black holes.
Pitfalls

1. Thresholding attacks
   a. Only use STAR for string measurement

2. Corrupt reports
   a. Verifiable STAR

3. Connection metadata
   a. Anonymizing proxy
Takeaways & goodbyes

1. Simplicity is important for user trust
2. Once data leaves device, all bets are off
3. Clarity of your unique requirements can lead to generally useful systems!
4. Boring crypto is good!
5. Academic/standards process can improve your system
6. Telemetry systems are surprisingly useful for many things!

Links: research paper, blog post, IETF draft
<table>
<thead>
<tr>
<th>Secret Sharing Scheme</th>
<th>Signature Scheme/Protocol</th>
<th>Client threat mitigated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shamir Secret Sharing</td>
<td>OPRF</td>
<td>None</td>
</tr>
<tr>
<td>Verifiable Secret Sharing</td>
<td>OPRF</td>
<td>Bad shares (DoS)</td>
</tr>
<tr>
<td>Shamir Secret Sharing</td>
<td>Blind Signatures</td>
<td>Bad ciphertext</td>
</tr>
<tr>
<td>Verifiable Secret Sharing</td>
<td>Blind Signatures</td>
<td>Both</td>
</tr>
</tbody>
</table>
Creating STAR

Cheap: low computational overhead and network usage
Simple
Private
Creating STAR

Cheap: low computational overhead and network usage
Simple: easy to implement, well-known crypto
Private
Creating STAR

Cheap: low computational overhead and network usage
Simple: easy to implement, well-known crypto
Private: practical privacy guarantees for clients