# CACTI: CAPTCHA Avoidance via Client-side TEE Integration

Yoshimichi Nakatsuka[1*], Ercan Ozturk[1*], Andrew Paverd[2], Gene Tsudik[1]

[1]*University of California, Irvine*

[2]*Microsoft Research*

**Primary Co-Authors*

# CAPTCHAs

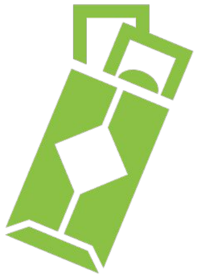
Completely Automated Public Turing test to tell Computers and Humans Apart

- Tasks that are “easy” to solve by humans, yet “difficult” for machines

“Any program that has high success over a captcha can be used to solve an unsolved Artificial Intelligence (AI) problem”

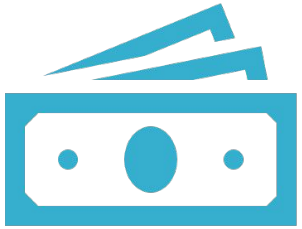
- von Ahn, et al. “CAPTCHA: Using Hard AI Problems for Security”, EUROCRYPT 2003

# Sample uses of CAPTCHAs



**Protecting high-demand and/or limited items**

Event ticket sales

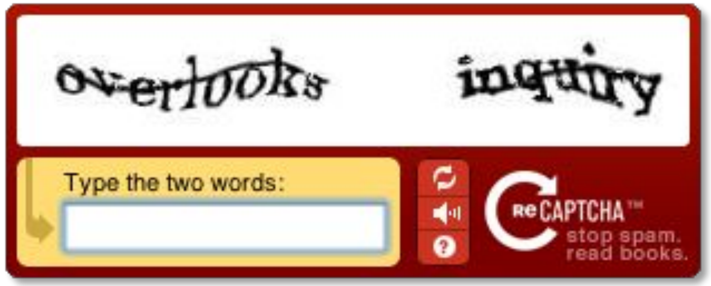


**Protecting high-value events**

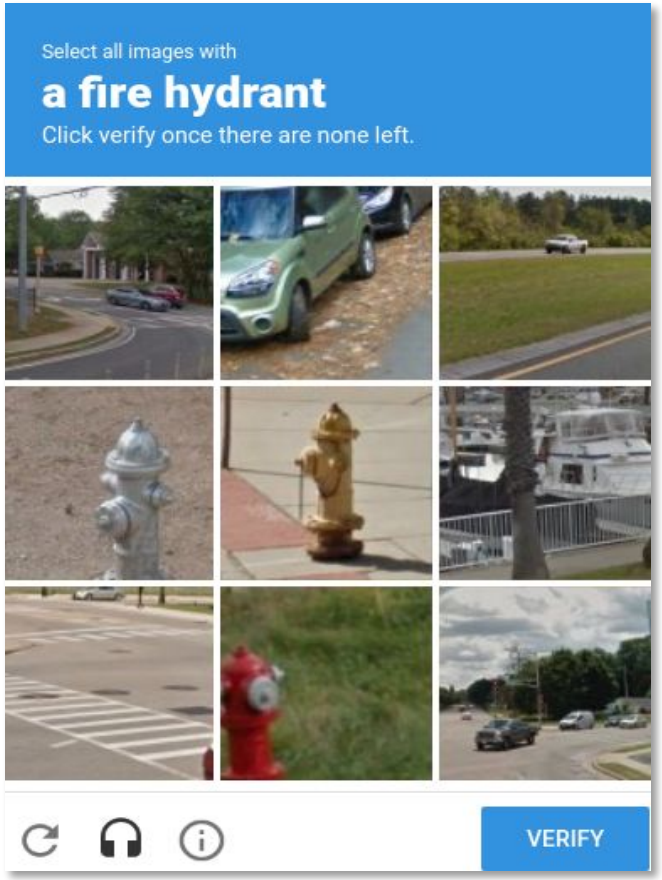
Signing up for a new account

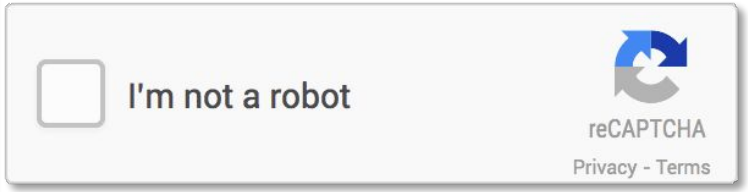
Voting in an online poll

# Common CAPTCHA Types




reCAPTCHA (http://www.captcha.net/)



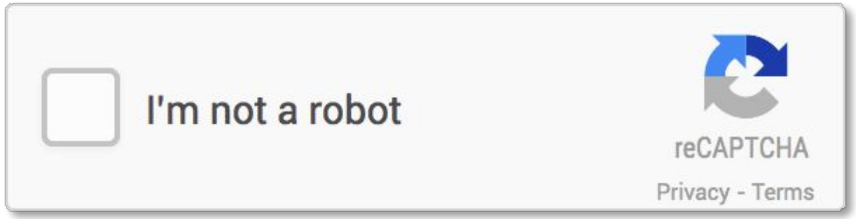



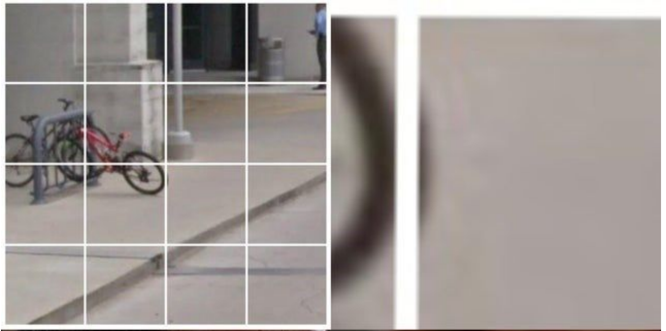

reCAPTCHA v2 (https://www.google.com/recaptcha/about/)

```
script
rc="https://www.google.com/recaptcha/api.js?render=
CAPTCHA_site_key"></script>

  <script>

  grecaptcha.ready(function() {

      grecaptcha.execute('reCAPTCHA_site_key',
action: 'homepage'}).then(function(token) { ...});

 });  </script>
```

reCAPTCHA v3 (https://www.google.com/recaptcha/about/)

# Downsides of CAPTCHAs

Difficult and time-consuming to solve

- Bursztein, et al. "[How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation]" IEEE Oakland 2010
- Fidas, et al. "[On the necessity of user-friendly CAPTCHA]" CHI 2011

Accessibility concerns

- Requires proper environment & device
- Blind or visually-impaired users?

Privacy concerns

- "[Google's new reCAPTCHA has a dark side]" Fast Company 2019
- "[Moving from reCAPTCHA to hCaptcha]" Cloudflare 2020

# Subverting CAPTCHAs

Modern machine learning can solve most types of CAPTCHAs

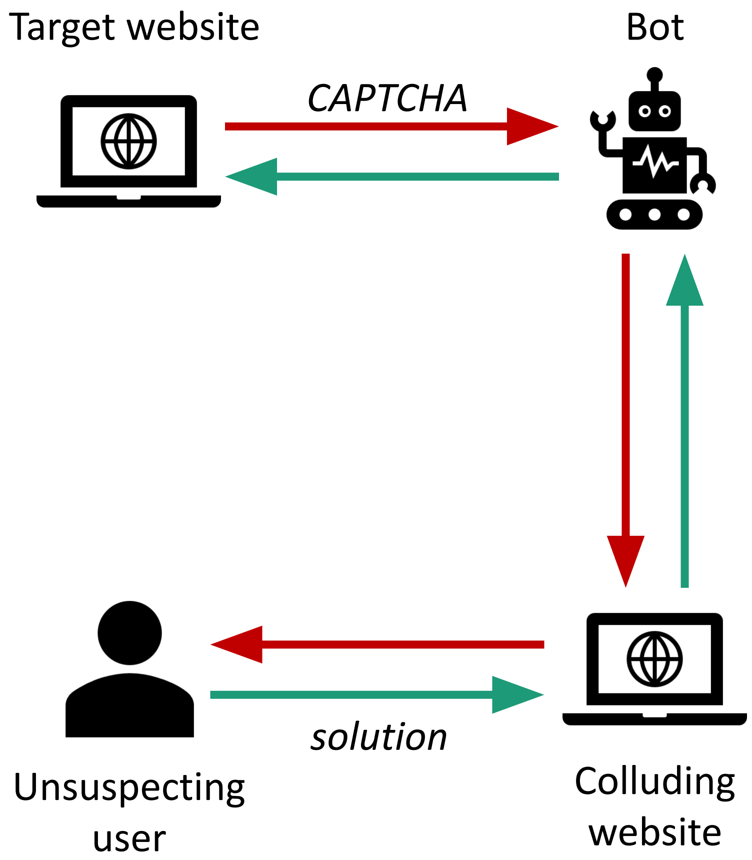
- Ye et al. "Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach" ACM CCS 2018
- Akrout et al. "Hacking Google reCAPTCHA v3 using Reinforcement Learning" arXiv 2019

CAPTCHA Forwarding attacks

- Bots forward CAPTCHAs to other sites to be solved by real users, e.g., discount coupon or porn sites

CAPTCHA farms

- Bots outsource CAPTCHA solving to human workers
- CAPTCHA solving services charge:
  - ~ $0.5 – $1 per 1000 CAPTCHAs
  - ~ $3 per 1000 reCAPTCHAs



# CAPTCHAs are still widely used

According to trends.builtwith.com (as of Feb 2021)

- reCAPTCHA ≈ 6.37 million live sites
- reCAPTCHA v3 ≈ 1.59 million live sites

**Claim:** Despite their drawbacks, CAPTCHAs are still used to:

- increase attacker costs (in terms of time or money) and/or
- reduce the rate of malicious activities

# Goals

- Minimize the number of CAPTCHAs shown to legitimate users without giving attackers a significant advantage

- Provide honest users a way to prove that they are not acting maliciously (for an appropriate definition of “maliciously”)

  - User: “I haven’t performed this action in the last *n* hours”
  - Website: “Prove it”
  - User: “OK, here’s a proof”

# CAPTCHA avoidance protocol

Instead of presenting a CAPTCHA, the following interaction takes place between the client (browser) and web server:

1) **Server** provides:
   - a rate threshold (i.e. # occurrences and a time period)
   - a timestamp for the current event

2) **Client** responds with proof that:
   - its rate for the specified action (within that time period) is below threshold, and
   - it has added the new timestamp to its database

# CAPTCHA avoidance protocol

Client maintains both **per-website** lists of timestamps and a **global list**

- Websites can specify which list to use for the rate-proof
- List ownership enforced through cryptographic signatures

Clients that cannot (or do not want to) provide a rate-proof simply fall back to being presented with a CAPTCHA

# Requirements and goals

Security

- Clients cannot forge or modify rate-proofs

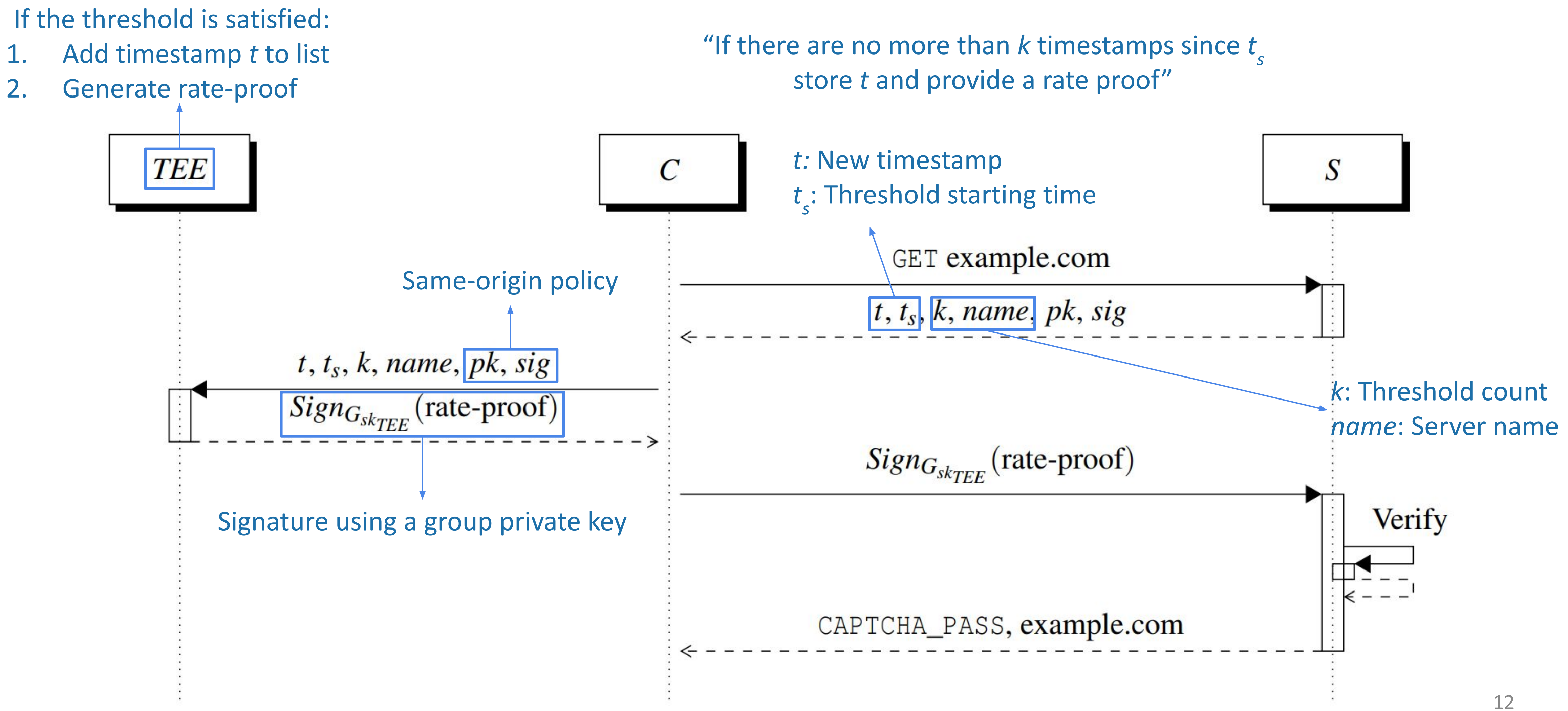
Privacy

- A server (or a group thereof) cannot link rate-proofs to the clients that generated them, or link two rate-proofs to the same client

Deployability

- Minimize user-perceived latency
- Minimize data transfer between client and server

# CAPTCHA avoidance using TEEs



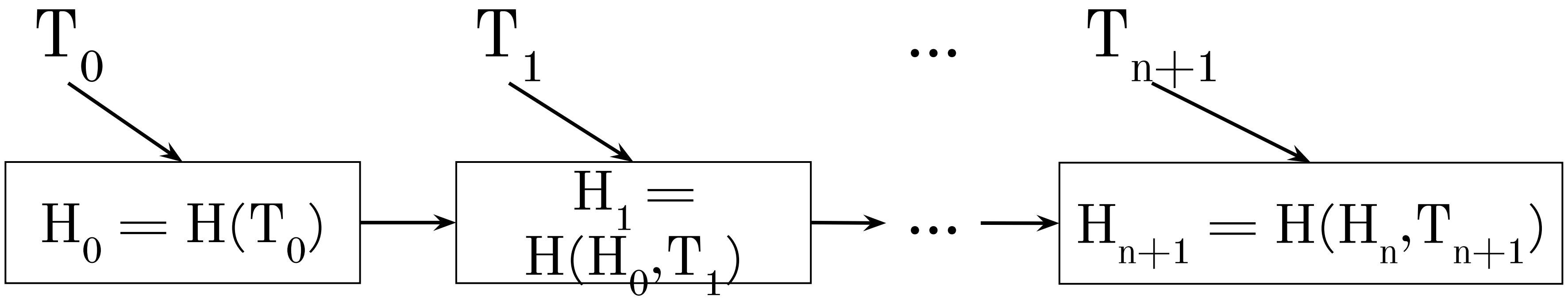
# Challenges & Solutions (1)

**Challenge**: limited amount of secure TEE memory

- e.g. current SGX enclaves have 100 MB

**Solution**: store timestamps outside the enclave but ensure integrity using hash chains

- only need integrity protection for the most recent hash



# Challenges & Solutions (2)

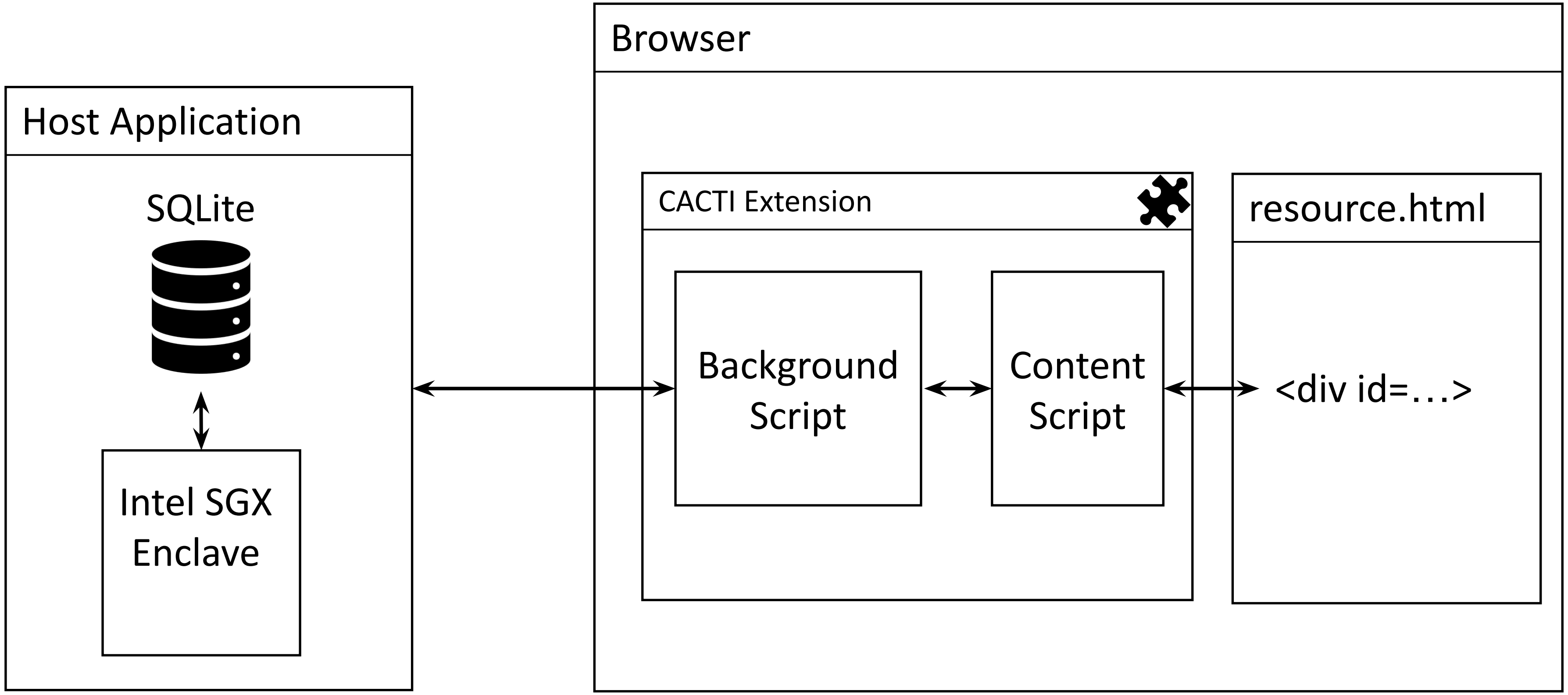
**Challenge**: limited number of monotonic counters

- e.g. current consumer SGX CPUs allow 256 counters per enclave

**Solution**: use a Merkle hash tree over the heads of the hash chains

- each leaf is the head of a website-specific hash-chain + list information
- only need roll-back protection for the root of the tree

# CACTI prototype implementation



# Security evaluation

| Adversarial client | Mitigated by |
|---|---|
| Data integrity & roll-back attacks | TEE security properties |
| Timestamp omission attacks | In-enclave checks |
| List substitution attacks | In-enclave checks |
| TEE reset attacks | Rate-limited by provisioning authority |
| TEE side-channel attacks | Ongoing research and/or new TEEs |
| CACTI Farms | Cost? |

# Security evaluation

| Adversarial server | Mitigated by |
|---|---|
| Client tracking | Group signature scheme<br>Not revealing actual client rates |

| Adversarial PA | Mitigated by |
|---|---|
| Does not verify remote attestation | Websites can decide which PAs to trust |

# Performance evaluation (Latency)

| | ECDSA-Sign | Browser | Pre-Enclave | In-Enclave | Post-Enclave | EPID-Verify | Total |
|---|---|---|---|---|---|---|---|
| 10,000 timestamps in 1 list | 6.3 ms | 15.2 ms | 7.7 ms | 181.7 ms | 1.0 ms | 27.3 ms | 239.2 ms |
| 4,096 lists with 1 timestamp each | 6.3 ms | 15.2 ms | 1.8 ms | 157.4 ms | 2.0 ms | 27.3 ms | 210.0 ms |

# Performance evaluation (Data transfer)

| | Received | Sent | Total |
|---|---|---|---|
| Image-based | 140.05 kB | 28.97 kB | 169.02 kB |
| Behavior-based | 54.38 kB | 26.12 kB | 80.50 kB |
| CACTI | 0.82 kB | 1.10 kB | 1.92 kB |

# CACTI: CAPTCHA Avoidance via Client-side TEE Integration

- Using client-side TEEs to provide signals of trustworthiness to websites
- Introduced CACTI rate-proof as a versatile primitive providing “express checkout” for legitimate users
- Proof-of-concept implementation within constraints of current TEE hardware
- Security level is server-configurable, and is always no worse than existing CAPTCHA schemes
- Possible enabler for new use cases?

**Questions?** [ercano@uci.edu](mailto:ercano@uci.edu) [nakatsuy@uci.edu](mailto:nakatsuy@uci.edu)