Unlock Web2 Data with TLSNotary

Hands-On Workshop

Sinu, Tsukino, Hendrik

Privacy + Scaling Explorations

Instructions https://tinyurl.com/tlsnotary

Overview

- **Introduction to TLSN**
- **Code Part I**
  - Solo
  - Local teams
- **Browser extension**
  - Demo
  - How it works
- **Code Part II**
  - Plugins
- **What is next?**
- **Playtime and Q&A**

Section 1

Introduction To TLSNotary

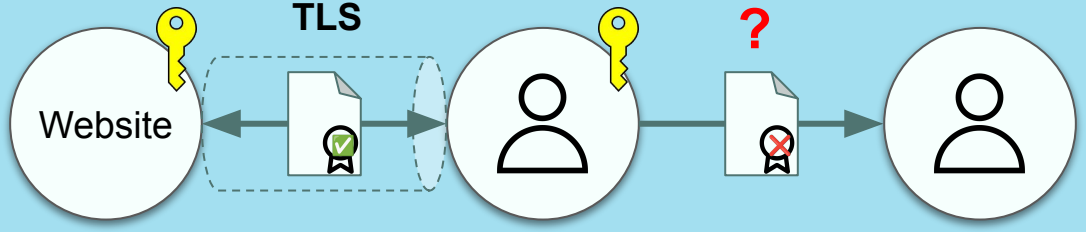

https://

Network Working Group
Request for Comments: 5246
Obsoletes: 3268, 4346, 4366
Updates: 4492
Category: Standards Track

T. Dierks
Independent
E. Rescorla
RTFM, Inc.

August 2008

The Transport Layer Security (TLS) Protocol Version 1.2


TLS
Website
?

MPC-TLS

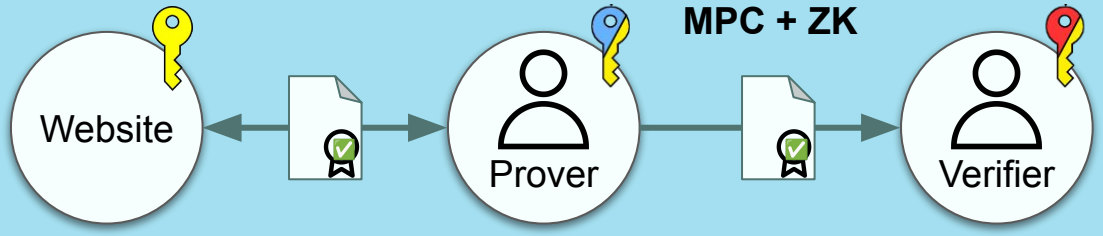
zkTLS

Web
Proof

Middlebox

Witness
Proxy

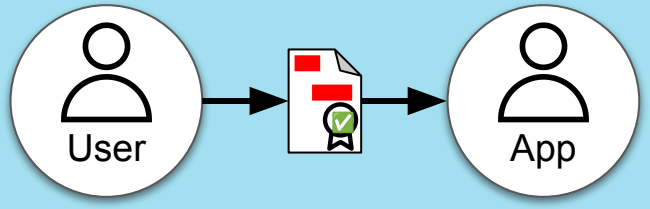
MPC-TLS



zkTLS (Proxy)

All approaches are **designated-verifier**

Compose with any application.
Trustlessly*.
Privately.

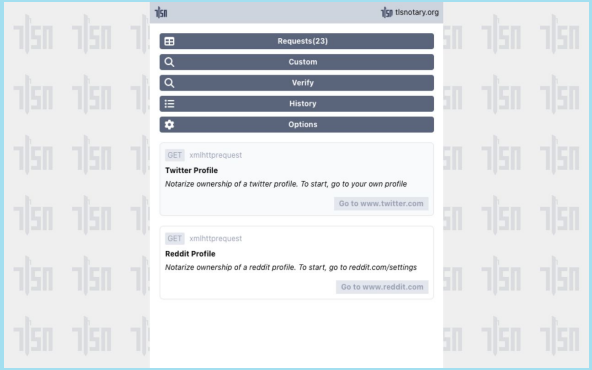



```
--- Request
GET /profile HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
Host: www.example.com
Accept-Language: en-us
Cookie: user_session=[redacted]
Connection: Keep-Alive

--- Response
HTTP/1.1 200 OK
Date: Wed, 14 Jun 2023 16:10:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Sat, 03 Jan 2009 19:15:56 GMT
Content-Length: 74
Content-Type: application/json
Connection: Closed

{"username":"sinu","address":"[redacted]","birthday":"[redacted]"}
```

Not in US ✅

18+ ✅

FOSS



License

All crates in this repository are licensed under either of

- Apache License, Version 2.0
- MIT license

at your option.

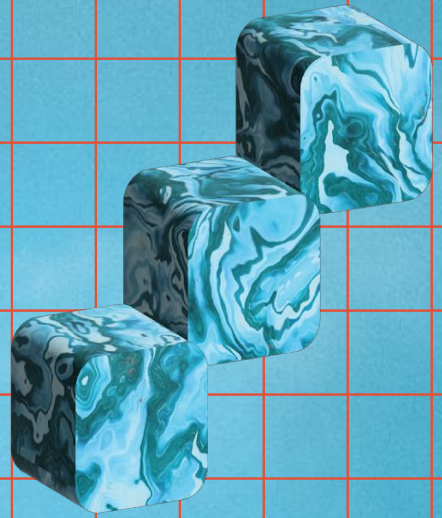
Section 2

Coding: Part I

Tips & Tricks

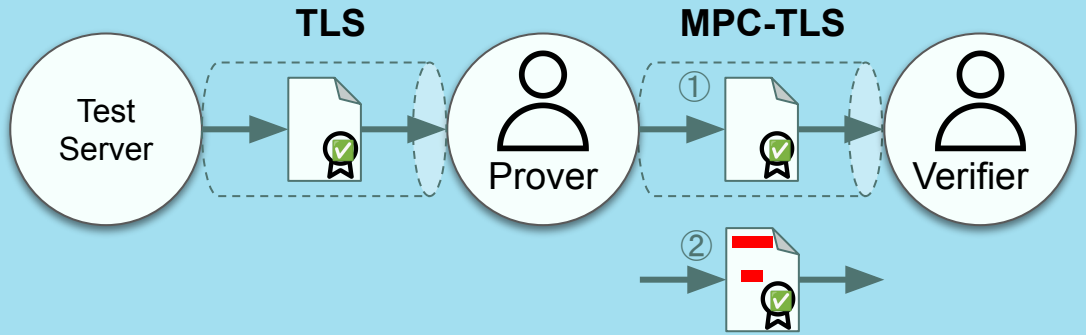
- We have time ⏳
- Comments 💬
- Wifi 📶
- Rust 🦀
- Ask questions ❓

Getting started



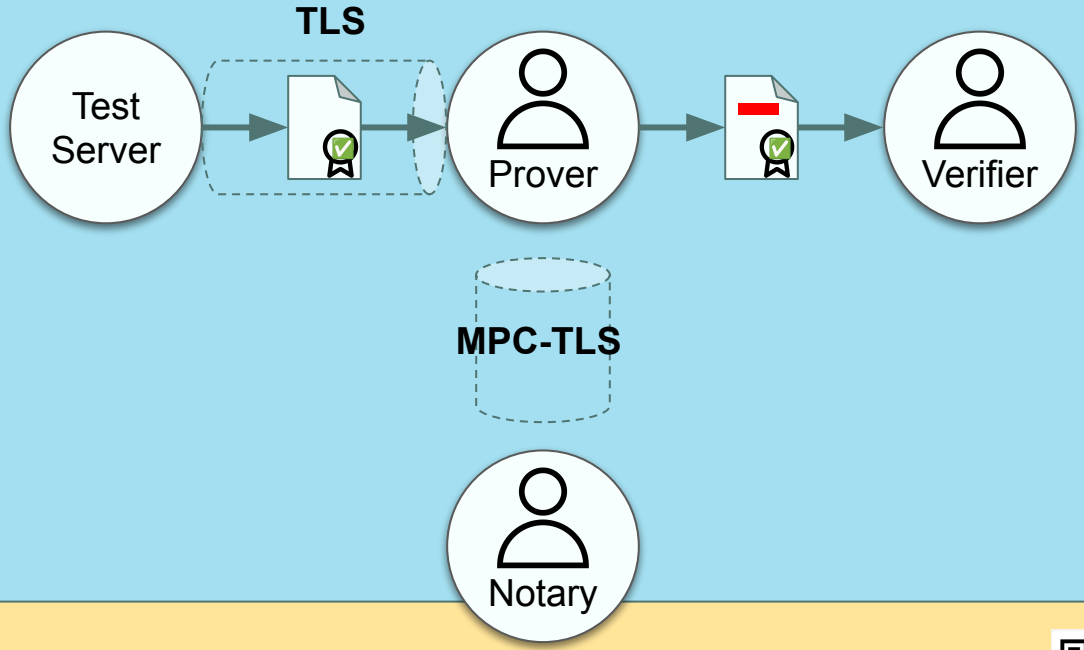
Instructions https://tinyurl.com/tlsnotary

1. Prove, redact and verify



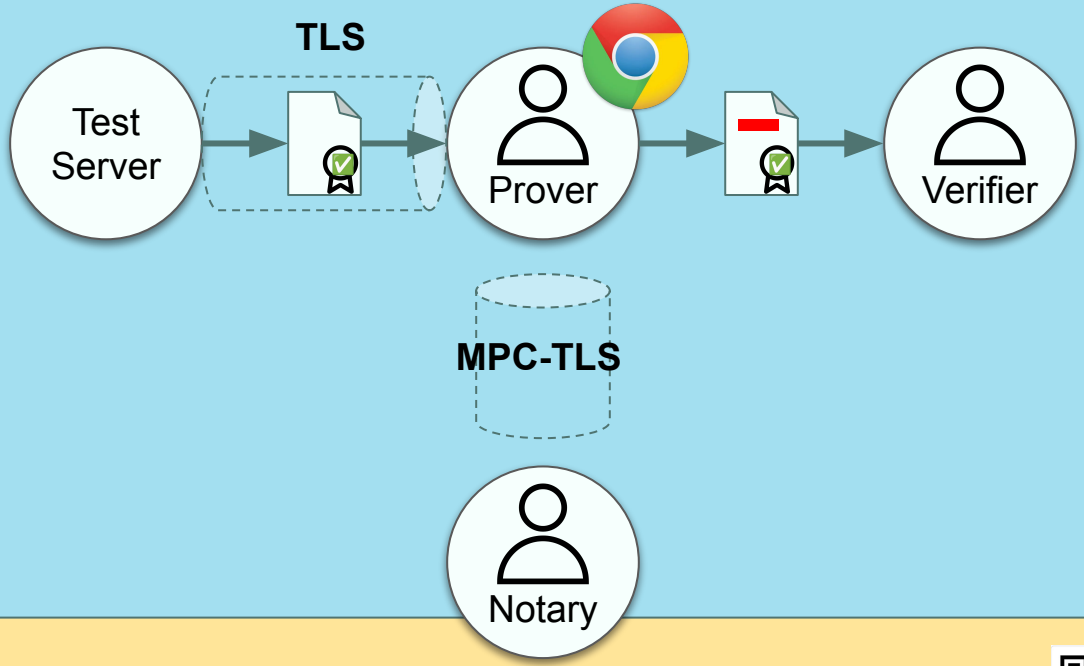
Instructions https://tinyurl.com/tlsnotary

2. Prove, redact and verify with a notary



Instructions https://tinyurl.com/tlsnotary

3. Prove, redact and verify in the Browser

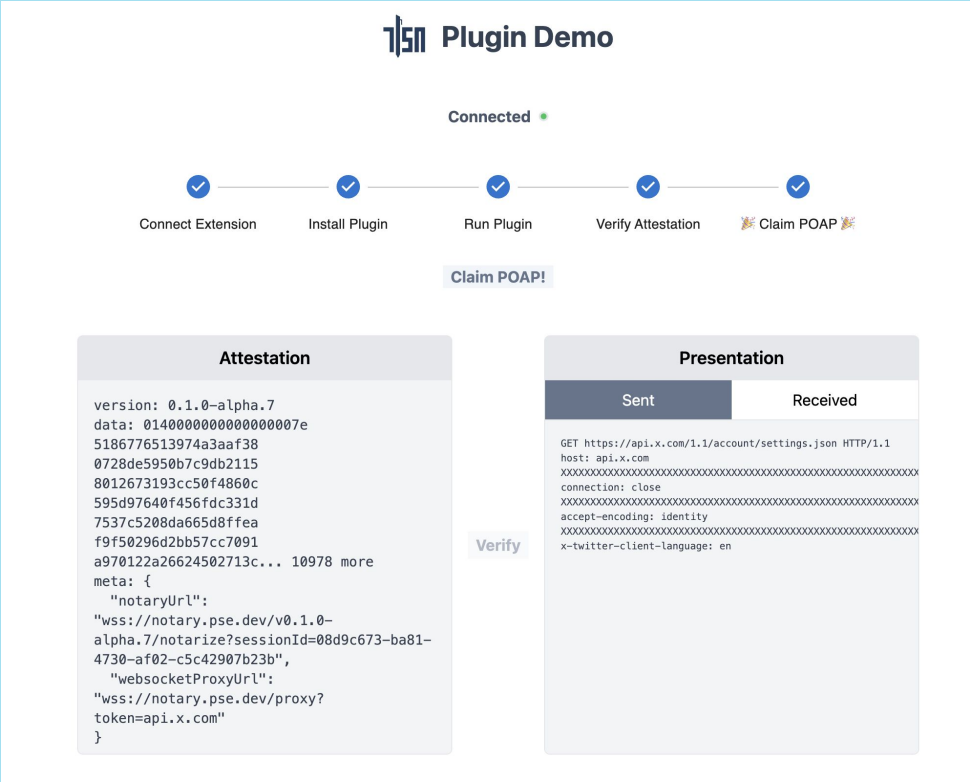


Instructions https://tinyurl.com/tlsnotary

Section 3

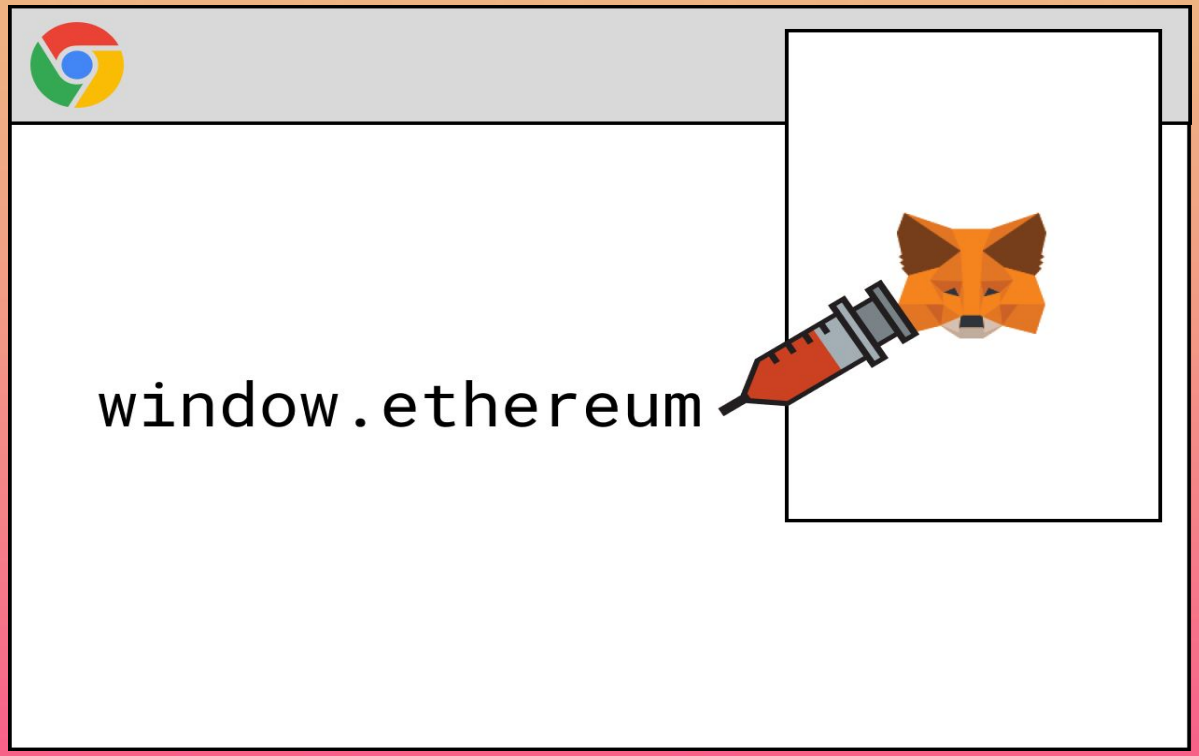
Browser Extension Connect API + Plugins

Web apps with TLSNotary: demo.tlsnotary.org

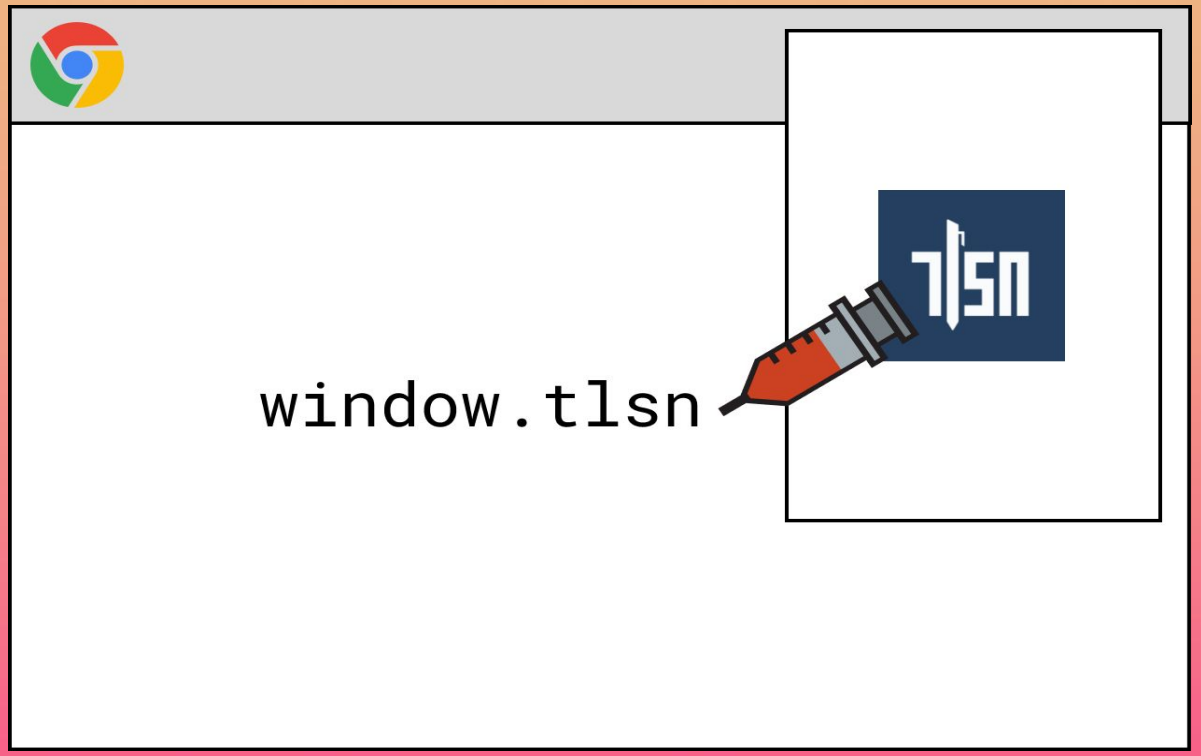


Connect API

docs.tlsnotary.org/extension/provider



docs.tlsnotary.org/extension/provider

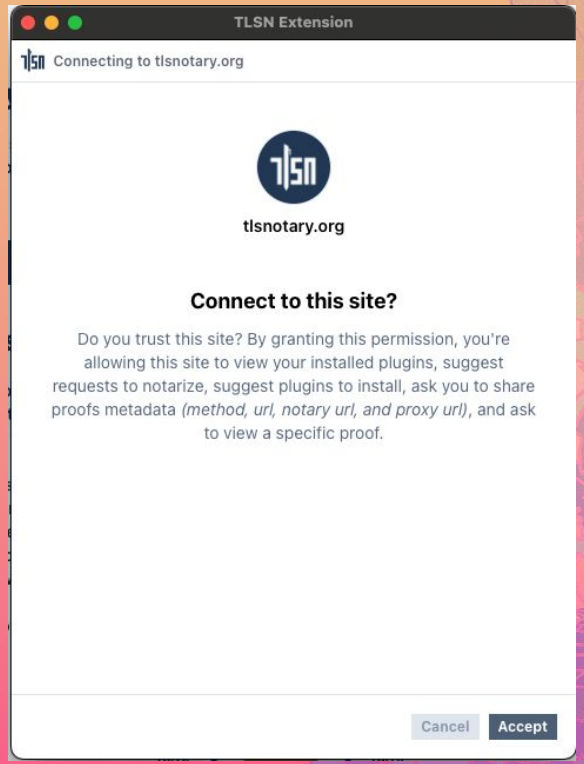


docs.tlsnotary.org/extension/provider

```
// 1. Wait for tlsn_loaded

window.addEventListener('tlsn_loaded', async () => {
    const tlsn = window.tlsn;
});
```

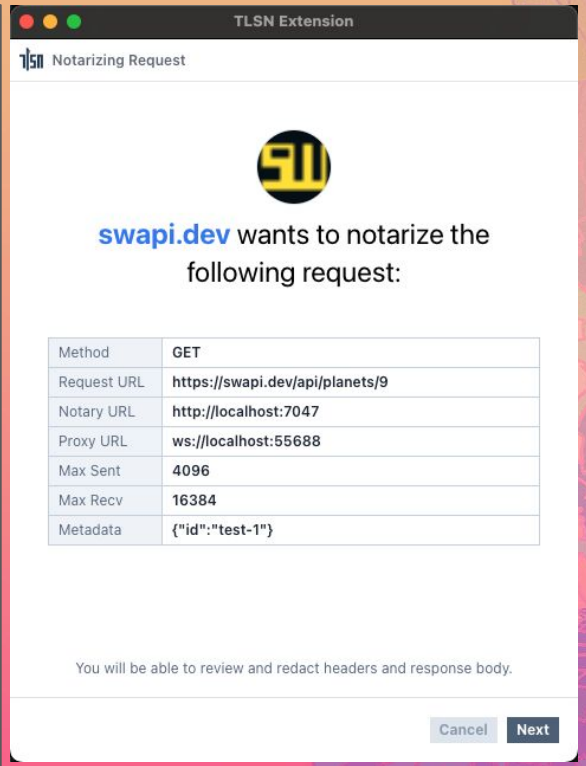
docs.tlsnotary.org/extension/provider

```
// 2. Connect to tlsn

window.addEventListener('tlsn_loaded', async () => {
    const tlsn = window.tlsn;
    const client = await tlsn.connect();
});
```



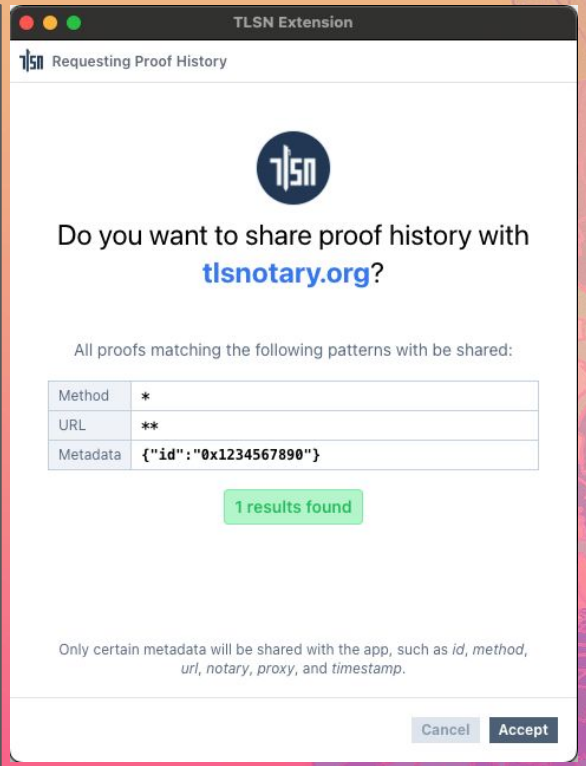
docs.tlsnotary.org/extension/provider

```
// 3. Notarization request

const proof = await client.notarize(
    'https://swapi.dev/api/planets/9',
    method: 'get',
    headers: {
        "Accept": "application/json",
        "Cookie": "csrftoken=blahblahblah",
    },
    metadata: {
        "id": "test-1",
    },
);
```



docs.tlsnotary.org/extension/provider

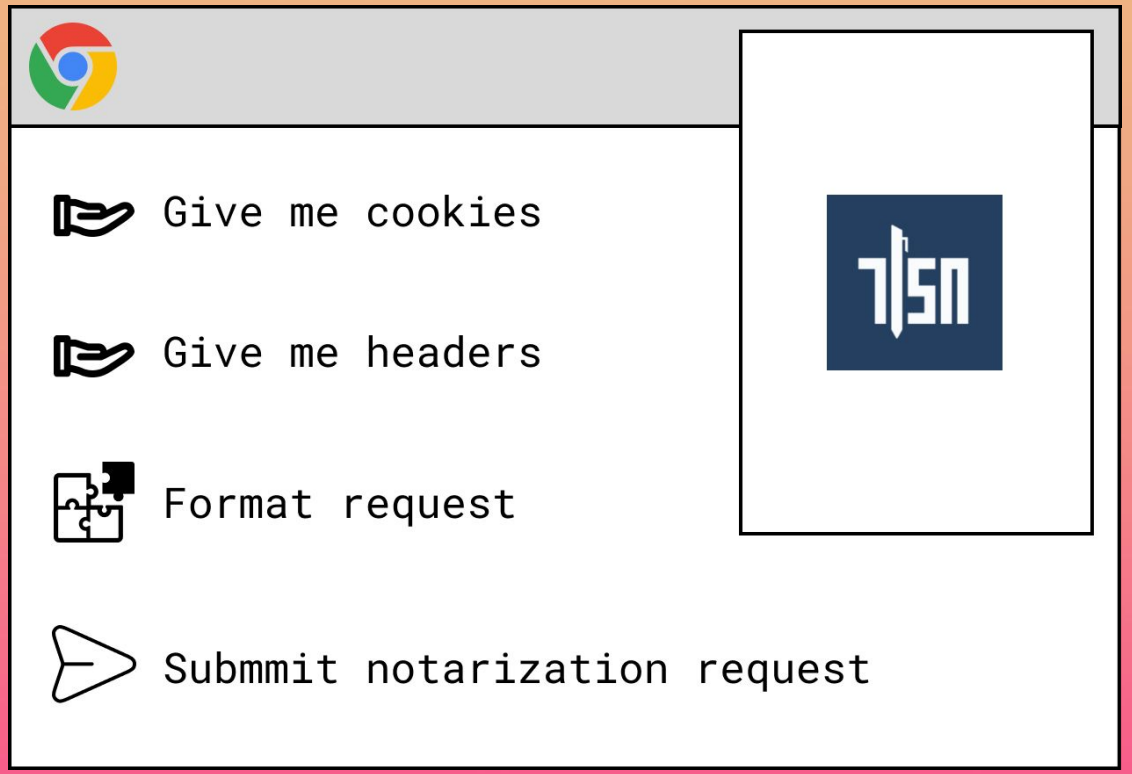
```
// 4. Ask for proof history

const results = await client.getHistory(
    'GET',
    'https://swapi.dev/api/plants/9',
);

// OR

const results = await client.getHistory(
    '*',
    '**',
    { id: 'test-1' },
);
```



docs.tlsnotary.org/extension/provider

Plugins

docs.tlsnotary.org/extension/plugins



docs.tlsnotary.org/extension/plugins

```
        "hostFunctions": [
            "redirect",
            "notarize"
        ],
        "cookies": [
            "api.x.com"
        ],
        "headers": [
            "api.x.com"
        ],
        "requests": [
            {
                "url": "https://api.x.com/1.1/account/settings.json",
                "method": "GET"
            }
        ]
    }
```

docs.tlsnotary.org/extension/plugins

```
{
    "title": "Twitter Profile",
    "description": "Notarize ownership of a twitter profile",
    "steps": [
        {
            "title": "Visit Twitter website",
            "cta": "Go to x.com",
            "action": "start"
        },
        {
            "title": "Collect credentials",
            "description": "Login to your account if you haven't already",
            "cta": "Check cookies",
            "action": "two"
        },
        {
            "title": "Notarize twitter profile",
            "cta": "Notarize",
            "action": "three",
            "prover": true
        }
    ],
```

docs.tlsnotary.org/extension/plugins

```
24    /**
25     * Implementation of the first (start) plugin step
26     */
27    export function start() {
28      if (!isValidHost(Config.get('tabUrl'))) {
29        redirect('https://x.com');
30        outputJSON(false);
31        return;
32      }
33      outputJSON(true);
34    }
```

docs.tlsnotary.org/extension/plugins

```
export function two() {
  const cookies = getCookiesByHost('api.x.com');
  const headers = getHeadersByHost('api.x.com');

  if (
    !cookies.auth_token ||
    !cookies.ct0 ||
    !headers['x-csrf-token'] ||
    !headers['authorization']
  ) {
    outputJSON(false);
    return;
  }

  outputJSON({
    url: 'https://api.x.com/1.1/account/settings.json',
    method: 'GET',
    headers: {
      'x-twitter-client-language': 'en',
      'x-csrf-token': headers['x-csrf-token'],
      Host: 'api.x.com',
      authorization: headers.authorization,
      Cookie: `lang=en; auth_token=${cookies.auth_token}; ct0=${cookies.ct0}`,
      'Accept-Encoding': 'identity',
      Connection: 'close',
    },
    secretHeaders: [
      `x-csrf-token: ${headers['x-csrf-token']}`,
      `cookie: lang=en; auth_token=${cookies.auth_token}; ct0=${cookies.ct0}`,
      `authorization: ${headers.authorization}`,
    ],
  });
}
```

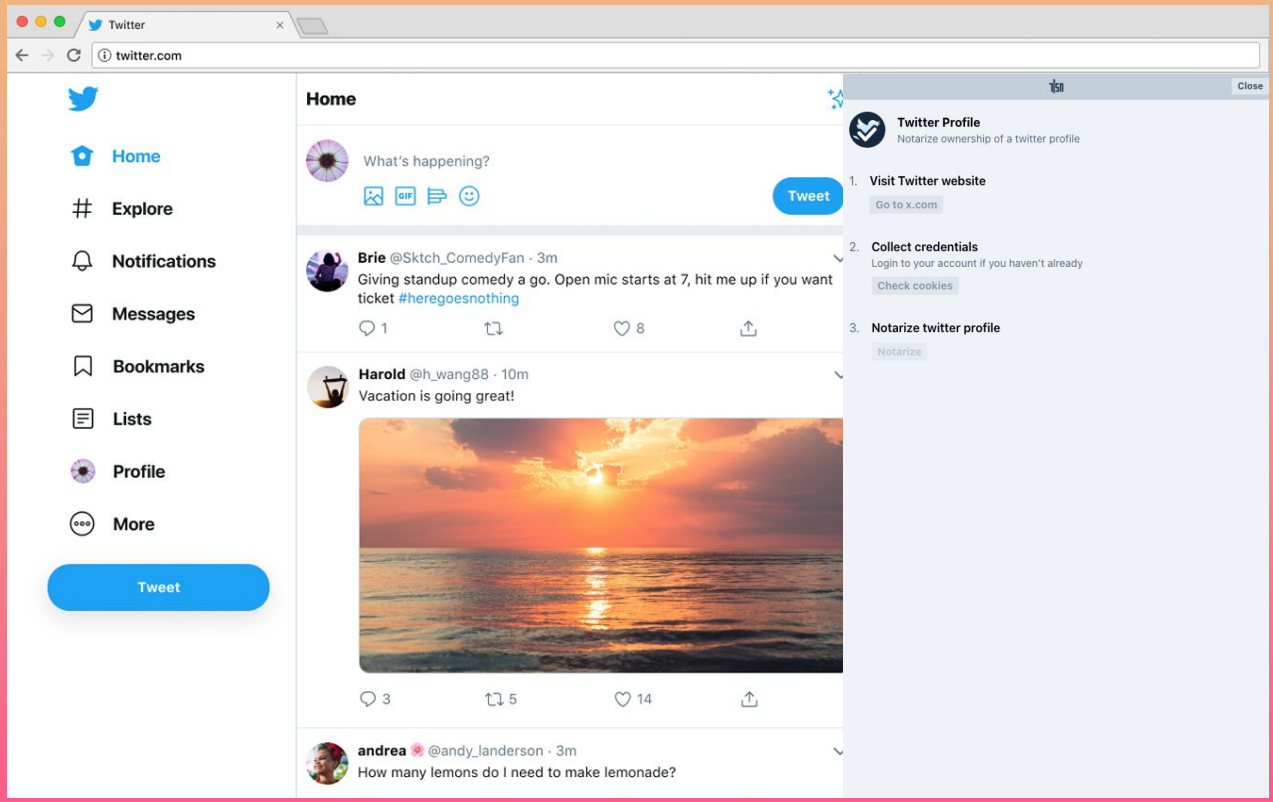
docs.tlsnotary.org/extension/plugins

```
104  export function three() {
105    const params = JSON.parse(Host.inputString());
106
107    if (!params) {
108      outputJSON(false);
109    } else {
110      const id = notarize({
111        ...params,
112        getSecretResponse: 'parseTwitterResp',
113      });
114      outputJSON(id);
115    }
116  }
```

docs.tlsnotary.org/extension/plugins

```
export function parseTwitterResp() {
  const bodyString = Host.inputString();
  const params = JSON.parse(bodyString);

  if (params.screen_name) {
    const revealed = `"screen_name":"${params.screen_name}"`;
    const selectionStart = bodyString.indexOf(revealed);
    const selectionEnd =
      selectionStart + revealed.length;
    const secretResps = [
      bodyString.substring(0, selectionStart),
      bodyString.substring(selectionEnd, bodyString.length),
    ];
    outputJSON(secretResps);
  } else {
    outputJSON(false);
  }
}
```

docs.tlsnotary.org/extension/plugins

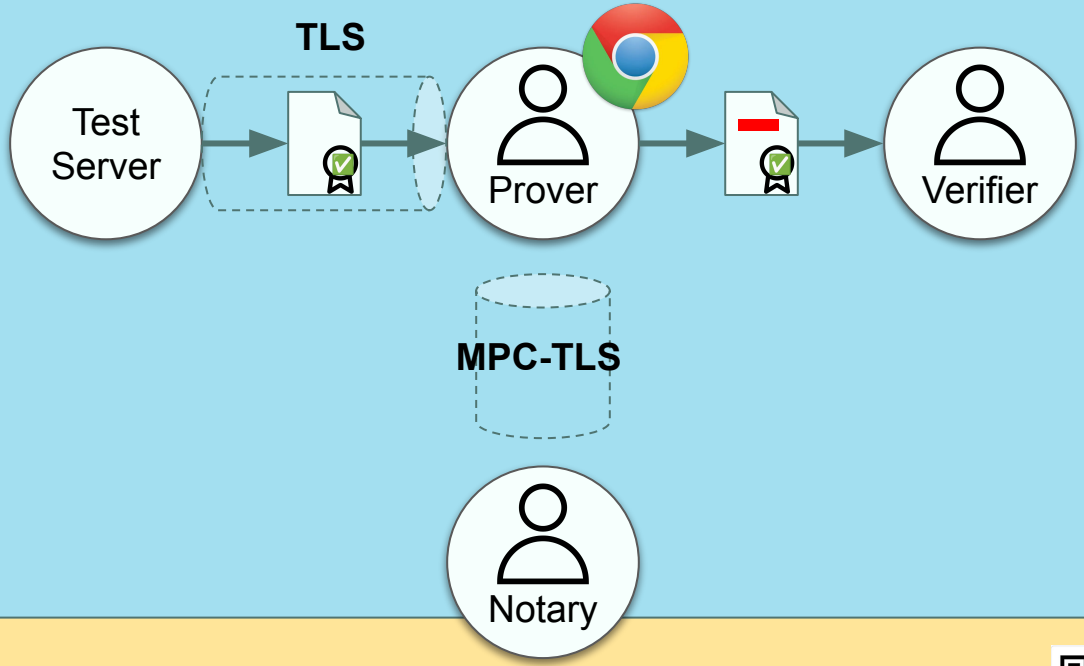


docs.tlsnotary.org/extension/plugins

Section 4

Coding: Part II

Build and test the Twitter profile plugin



Instructions https://tinyurl.com/tlsnotary

Section 5

TLSNotary
What is next?

Alpha.8 Soon™

QuickSilver 🏎️

AuthDecode 🧩

Benchmarks 📈(🦀/🌐)

TLS 1.3

Audit

Section 6

Playtime and Q&A

Thank you!

TLSNotary team

PSE

https://tlsnotary.org/