I Block You Because I Love You:

Social Account Identification Attack Against a Website Visitor

Takuya Watanabe

NTT Secure Platform Laboratories

black hat EUROPE 2018

DECEMBER 3-6, 2018
EXCEL LONDON, UK

WWW.BLACKHAT.COM #BHEU

Who am I?

◆ Takuya Watanabe

- NTT Secure Platform Laboratories, Japan
- Ph.D. student at Waseda University
- Interests: Web Sec. / Mobile Sec. / Side-channel Attack / Consumer Privacy
- E-mail: watanabe.takuya@lab.ntt.co.jp
- Twitter: @twatanabe1203

◆ Co-authors

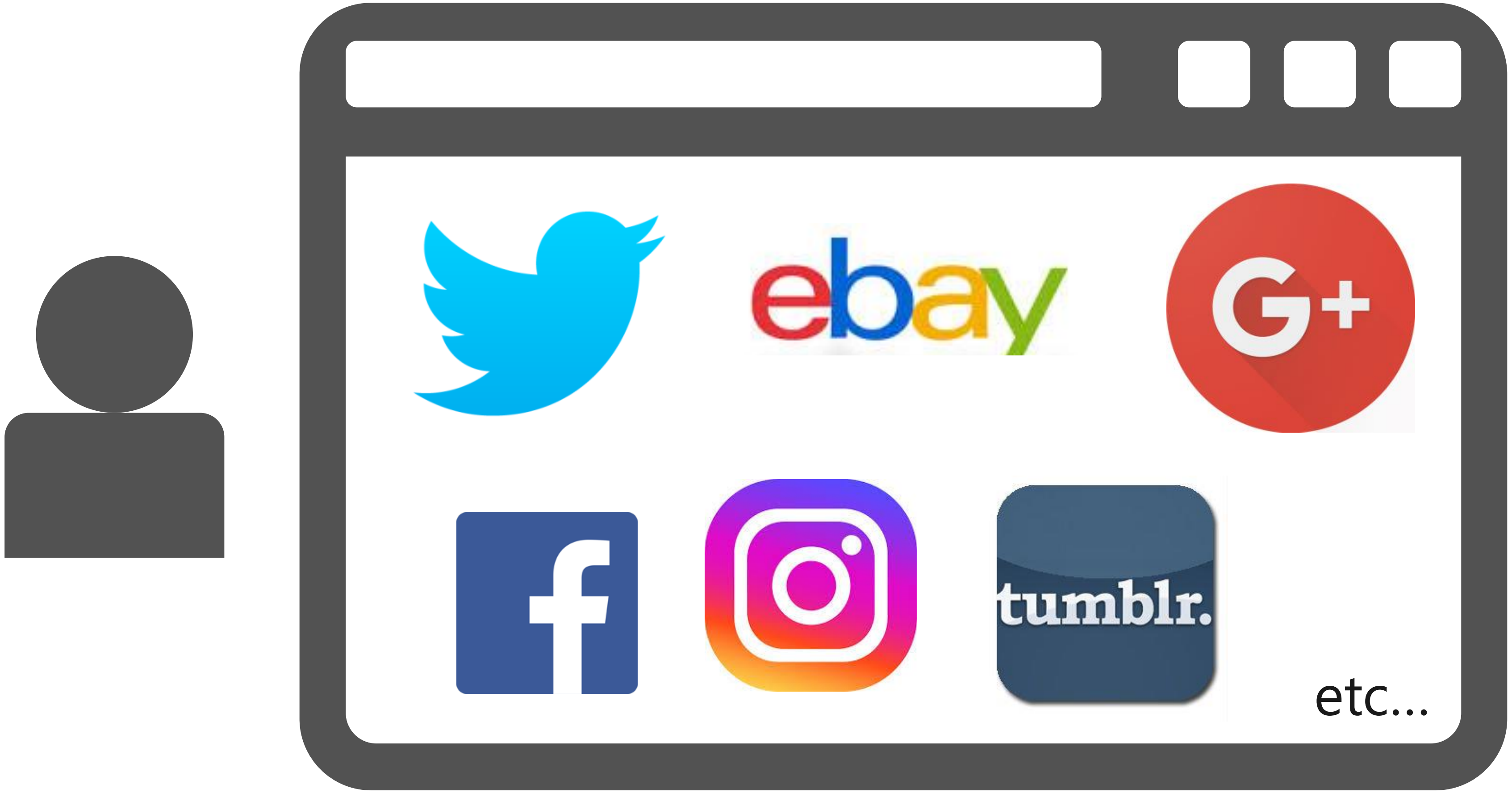
- Eitaro Shioji (NTT Secure Platform Labs.)
- Mitsuaki Akiyama (NTT Secure Platform Labs.)
- Keito Sasaoka (Waseda University)
- Takeshi Yagi (NTT Security Japan)
- Tatsuya Mori (Waseda University)

About this research

- Privacy threat called "Silhouette“
  - Our press release:
    http://www.ntt.co.jp/news2018/1807e/180718a.html
  - Twitter's writeup:
    https://blog.twitter.com/engineering/en_us/topics/insights/2018/twitter_silhouette.html
    (or https://t.co/0BQ59NuZ0V)

- Research Impact
  - Bring up new security problem
  - Remediation of major social web services
  - Support of the SameSite attribute by major browsers

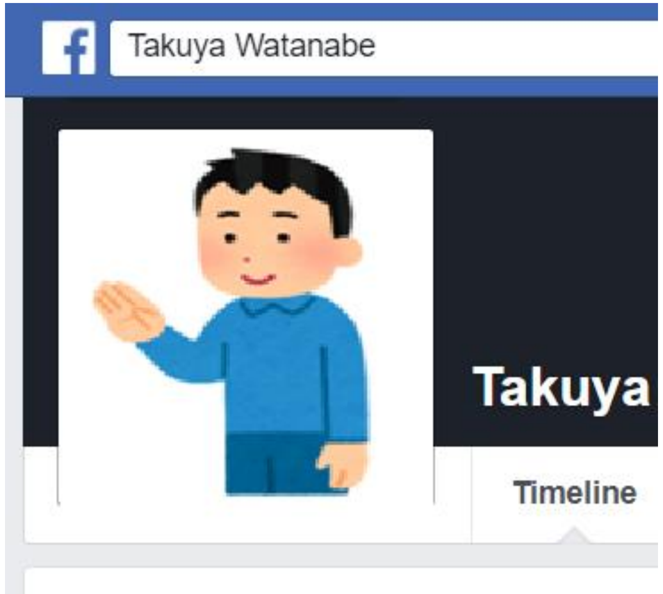
Widespread Adoption of Social Webs



Internet users have an average of 5+ social accounts

Social Accounts Contain…

- Personal information
  - Real name
  - Photo
  - Location

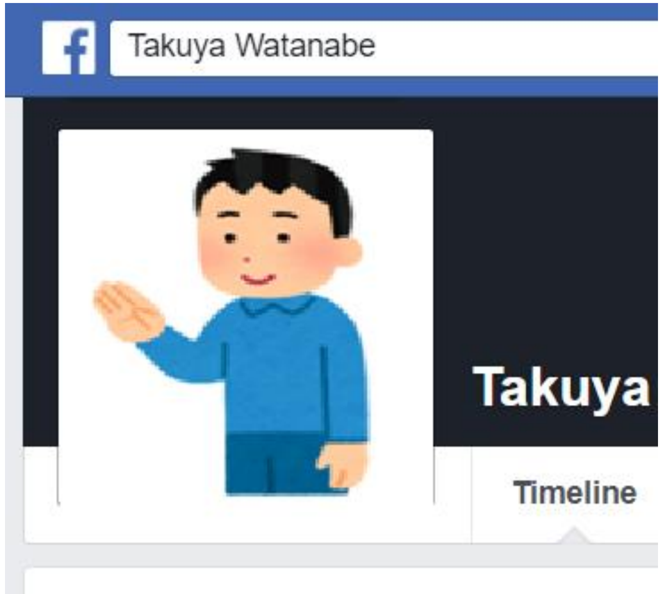


Takuya Watanabe

Takuya

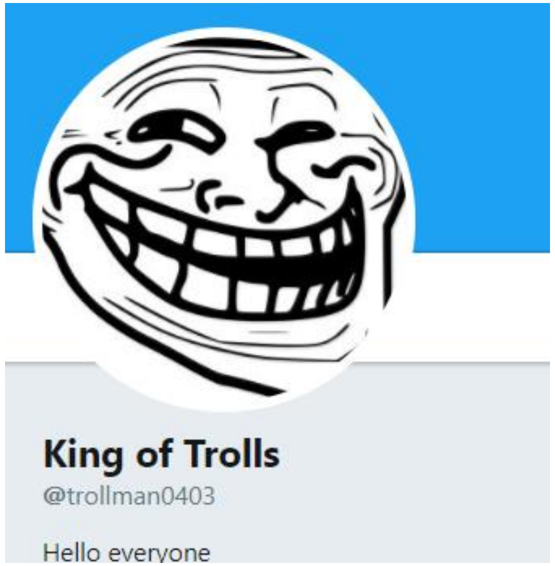
Social Accounts Contain...

- Personal information
  - Real name
  - Photo
  - Location



Takuya Watanabe

Takuya



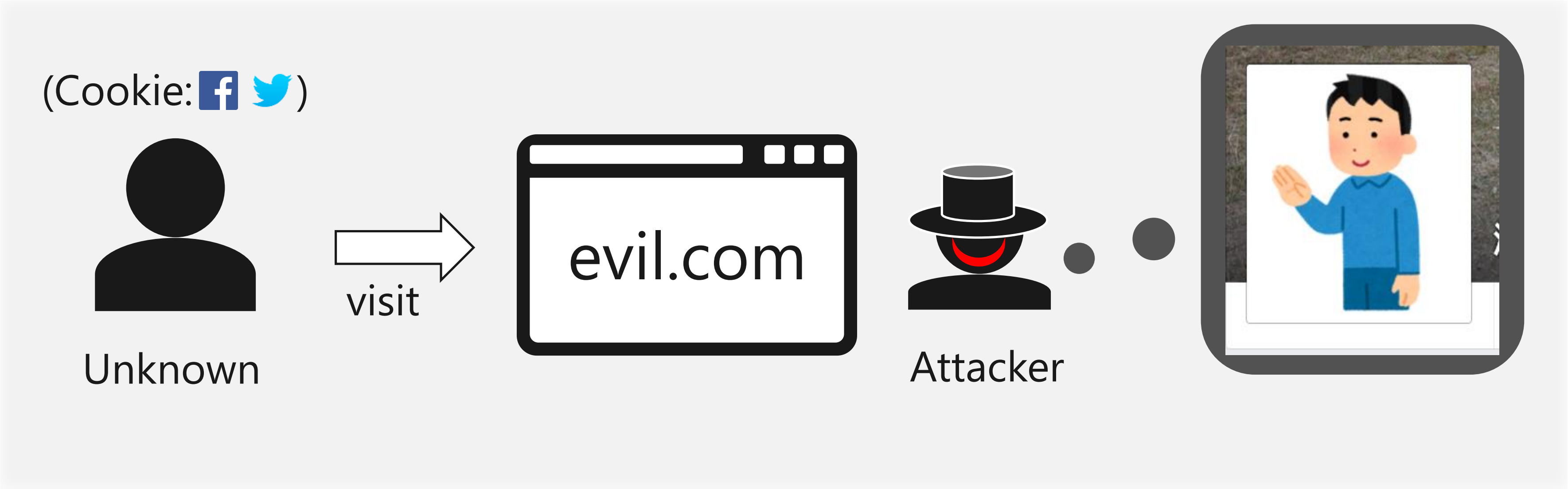
King of Trolls

@trollman0403

Hello everyone

- Secret activities
  - Screen name
  - Purchase history
  - Use of porn or dating sites

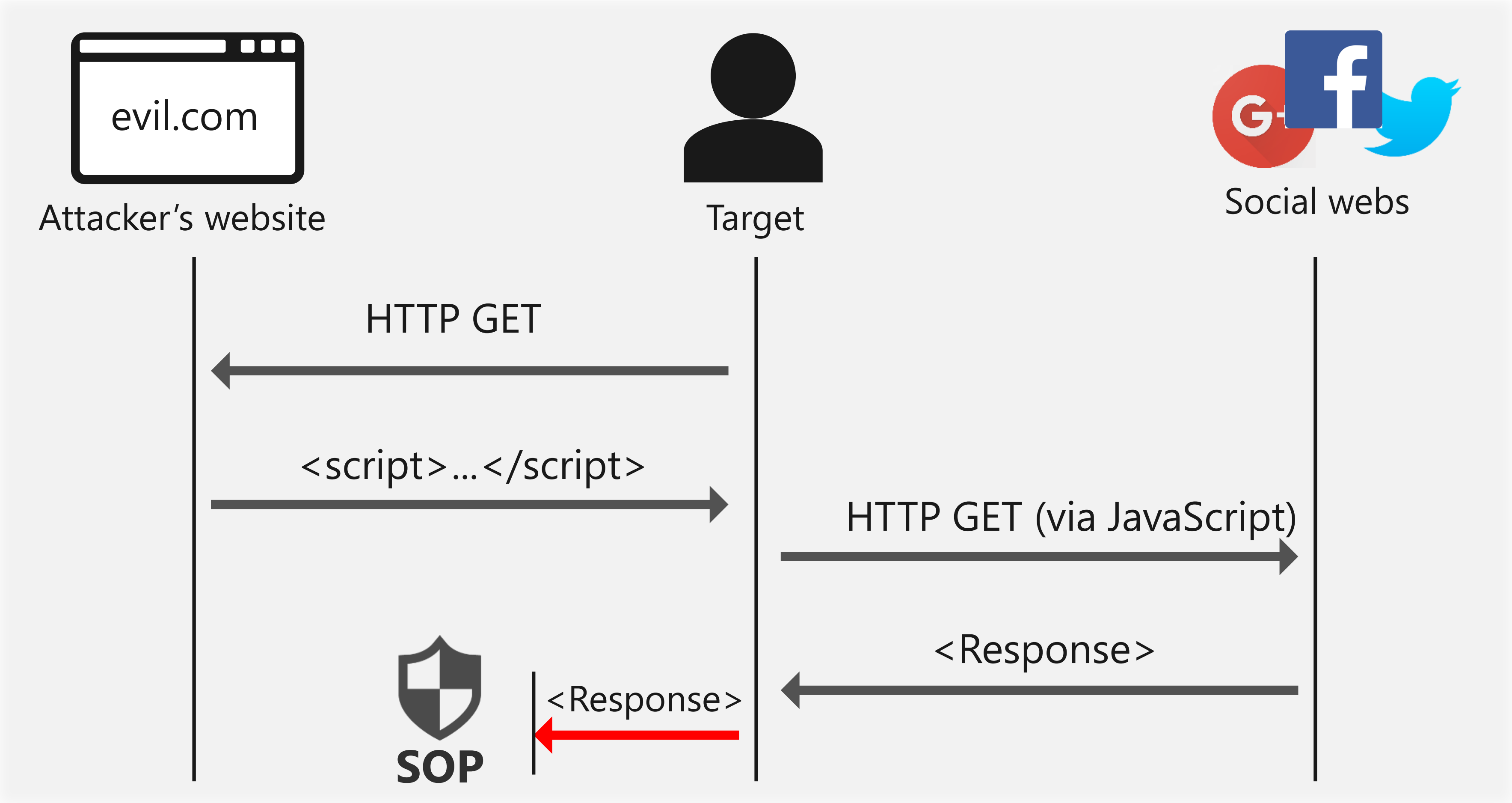
Threat Model: Social Account Identification



- The anonymity of a website visitor can be destroyed by identifying the social account.
- It allows
  - Tracking and stalking
  - Social engineering
  - Blackmailing
  - …

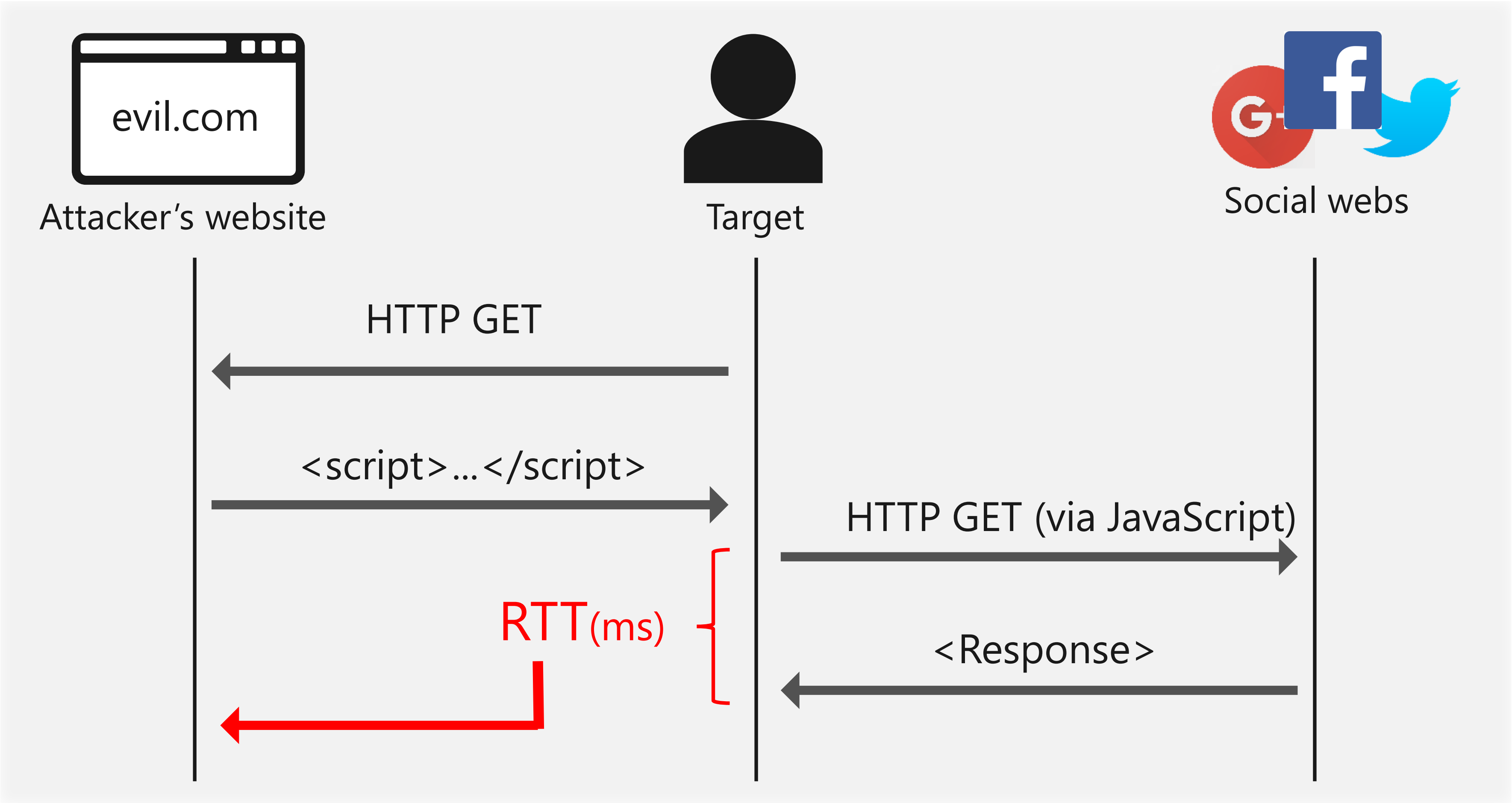
Technical Background

Same Origin Policy



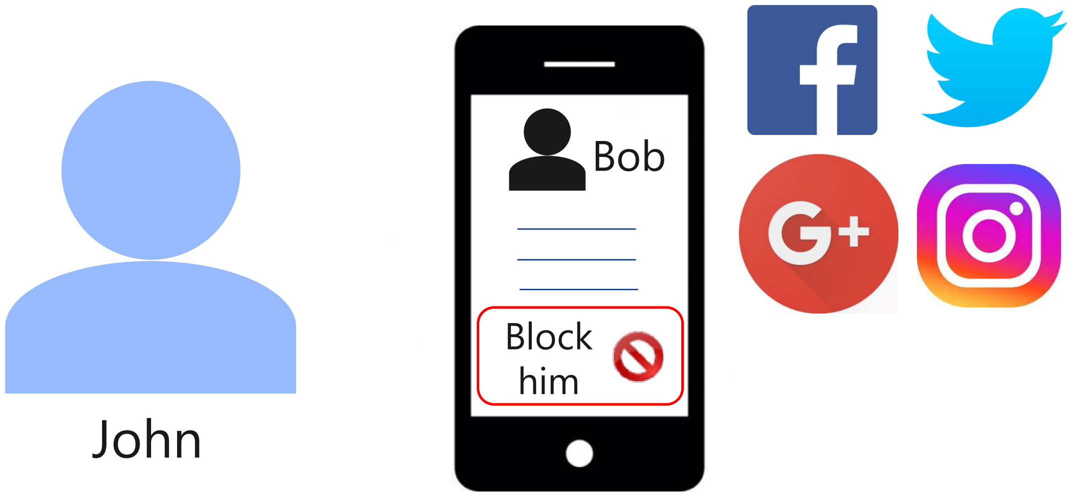
Cross-site responses are protected by SOP

Same Origin Policy



<u>The required time (i.e. RTT) can be measured</u>

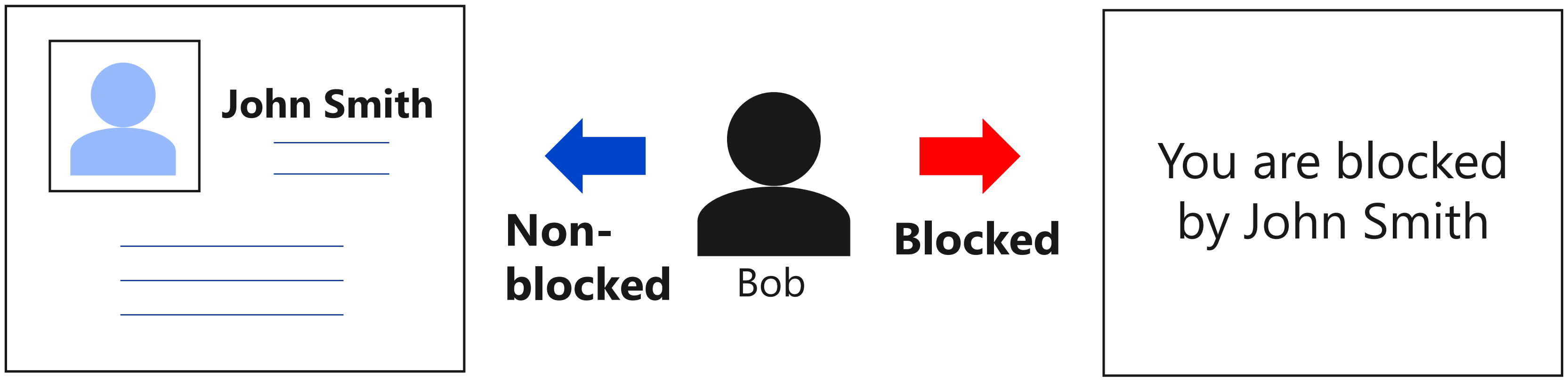
Key Idea: Visibility Control by User Blocking



John

Bob

Block him



John Smith

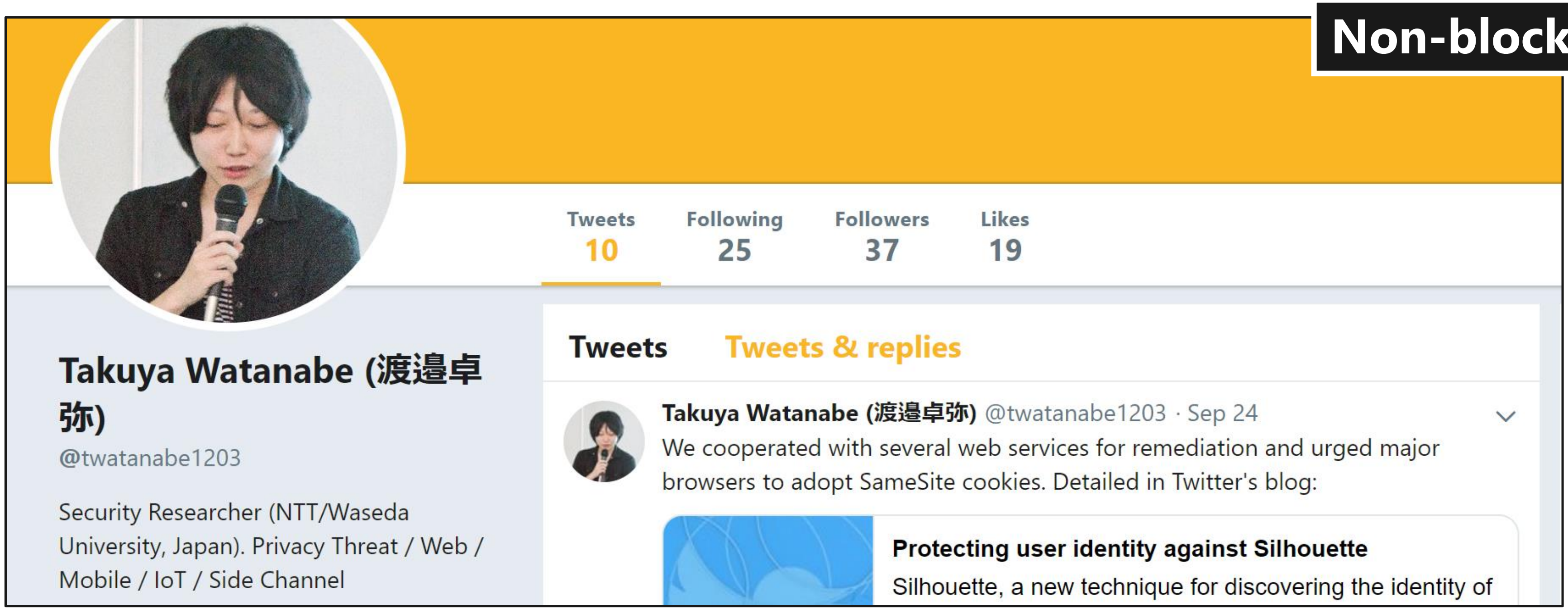
Non-blocked

Bob

Blocked

You are blocked by John Smith

Non-blocked



Tweets 10 | Following 25 | Followers 37 | Likes 19

Takuya Watanabe (渡邉卓弥)

@twatanabe1203

Security Researcher (NTT/Waseda University, Japan). Privacy Threat / Web / Mobile / IoT / Side Channel

Tweets | Tweets & replies

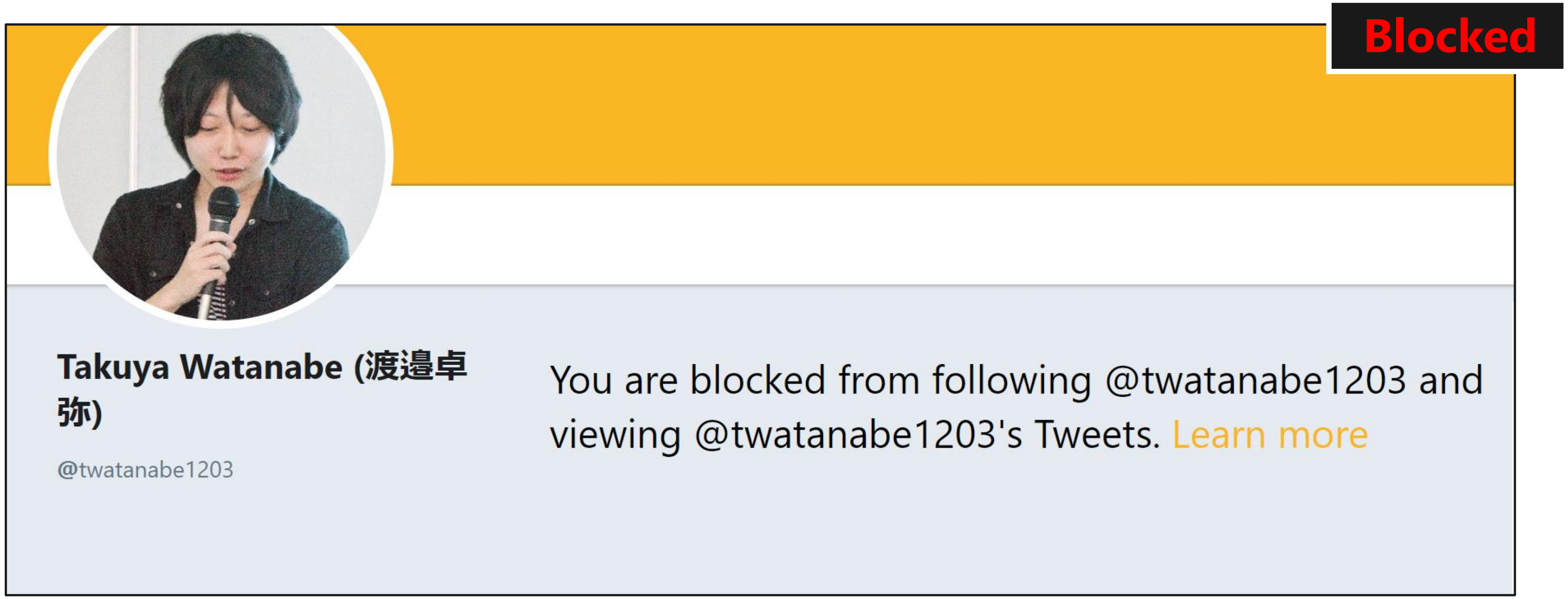
Takuya Watanabe (渡邉卓弥) @twatanabe1203 · Sep 24

We cooperated with several web services for remediation and urged major browsers to adopt SameSite cookies. Detailed in Twitter's blog:

Protecting user identity against Silhouette

Silhouette, a new technique for discovering the identity of

Blocked

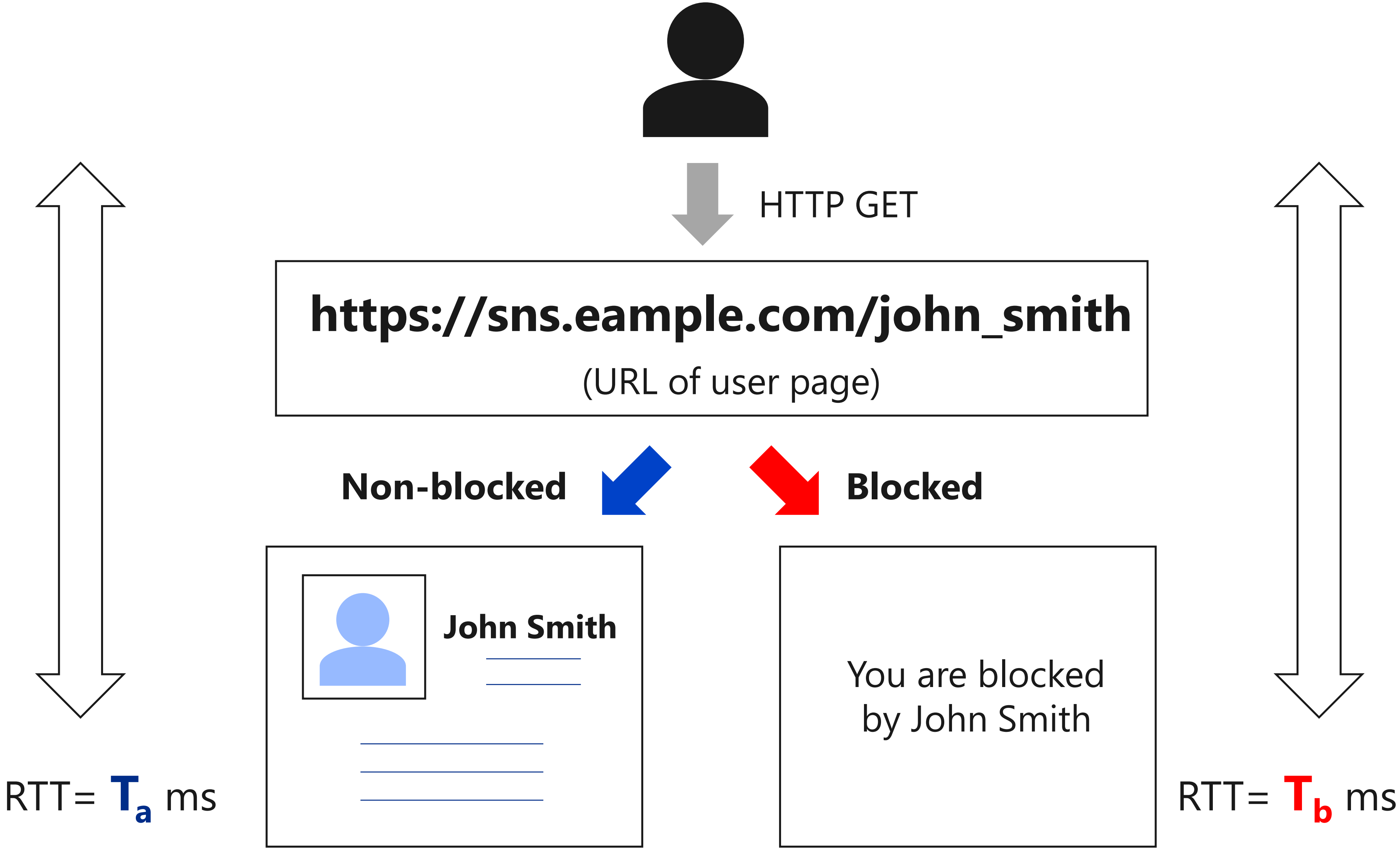


Takuya Watanabe (渡邉卓弥)

@twatanabe1203

You are blocked from following @twatanabe1203 and viewing @twatanabe1203's Tweets. Learn more

Key Idea: Visibility Control by User Blocking



HTTP GET

https://sns.eample.com/john_smith

(URL of user page)

Non-blocked

Blocked

John Smith

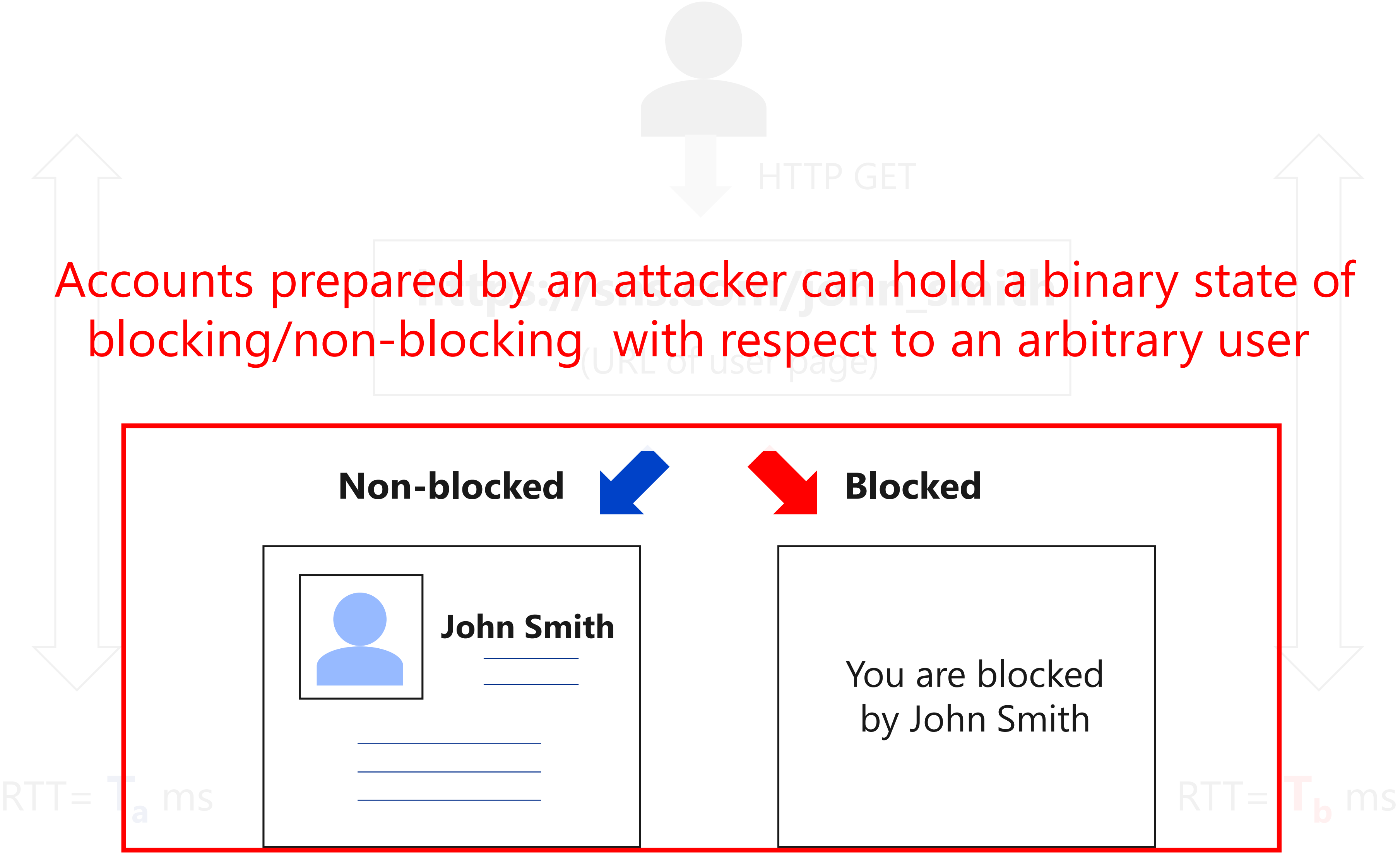
You are blocked by John Smith

RTT= T_a ms

RTT= T_b ms

Key Idea: Visibility Control by User Blocking

Accounts prepared by an attacker can hold a binary state of blocking/non-blocking with respect to an arbitrary user



You are blocked by John Smith

User Identification Attack

Attack Flow

I. Side-Channel Control Phase

To construct user-identifiable side-channel data through user blocking feature

➡ Required just once before performing the attack

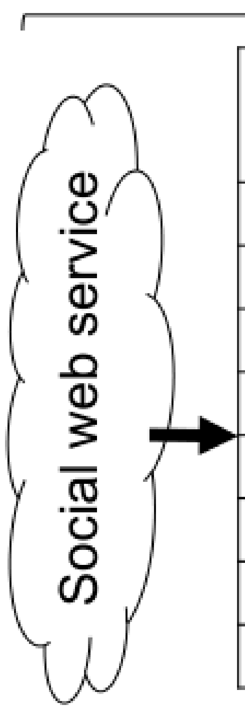
II. Side-Channel Retrieval Phase

To identify the user accounts utilizing the data retrieved through the timing side channel

➡ Executed every time a user accesses the attacker's website

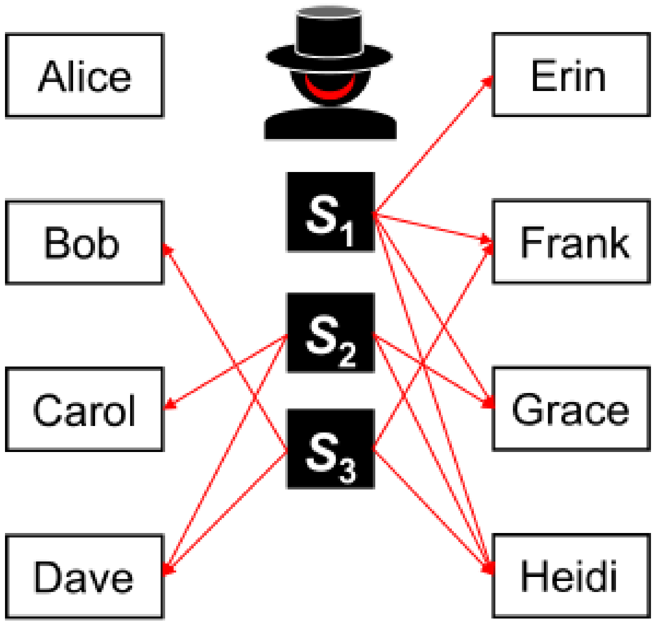
Side-Channel Control Phase

Step 1: Target Enumeration



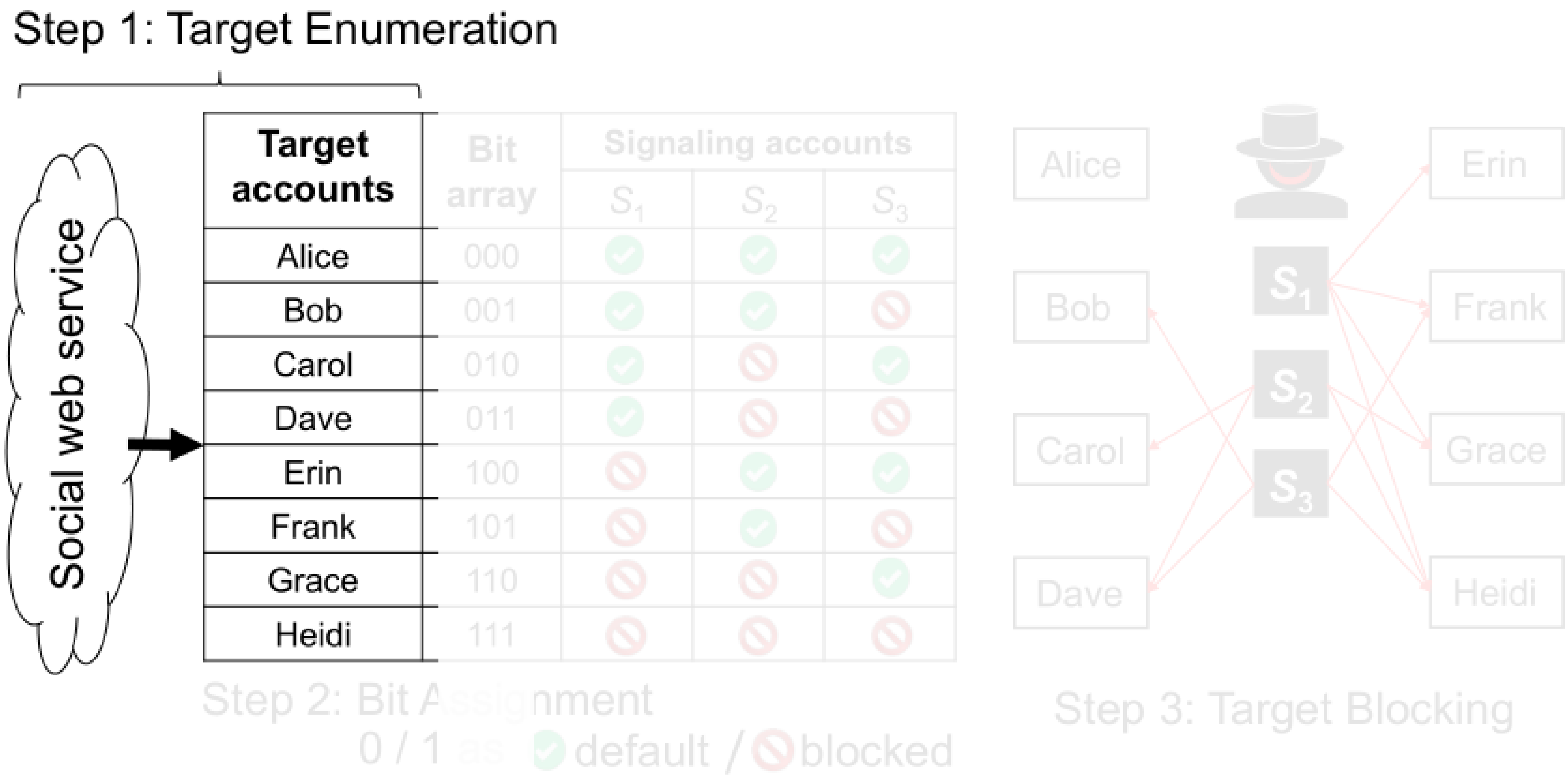
Target accounts	Bit array	Signaling accounts		
		S_1	S_2	S_3
Alice	000	✓	✓	✓
Bob	001	✓	✓	🚫
Carol	010	✓	🚫	✓
Dave	011	✓	🚫	🚫
Erin	100	🚫	✓	✓
Frank	101	🚫	✓	🚫
Grace	110	🚫	🚫	✓
Heidi	111	🚫	🚫	🚫

Step 2: Bit Assignment
0 / 1 as ✓ default / 🚫 blocked



Step 3: Target Blocking

Side-Channel Control Phase



Step 1: Target Enumeration

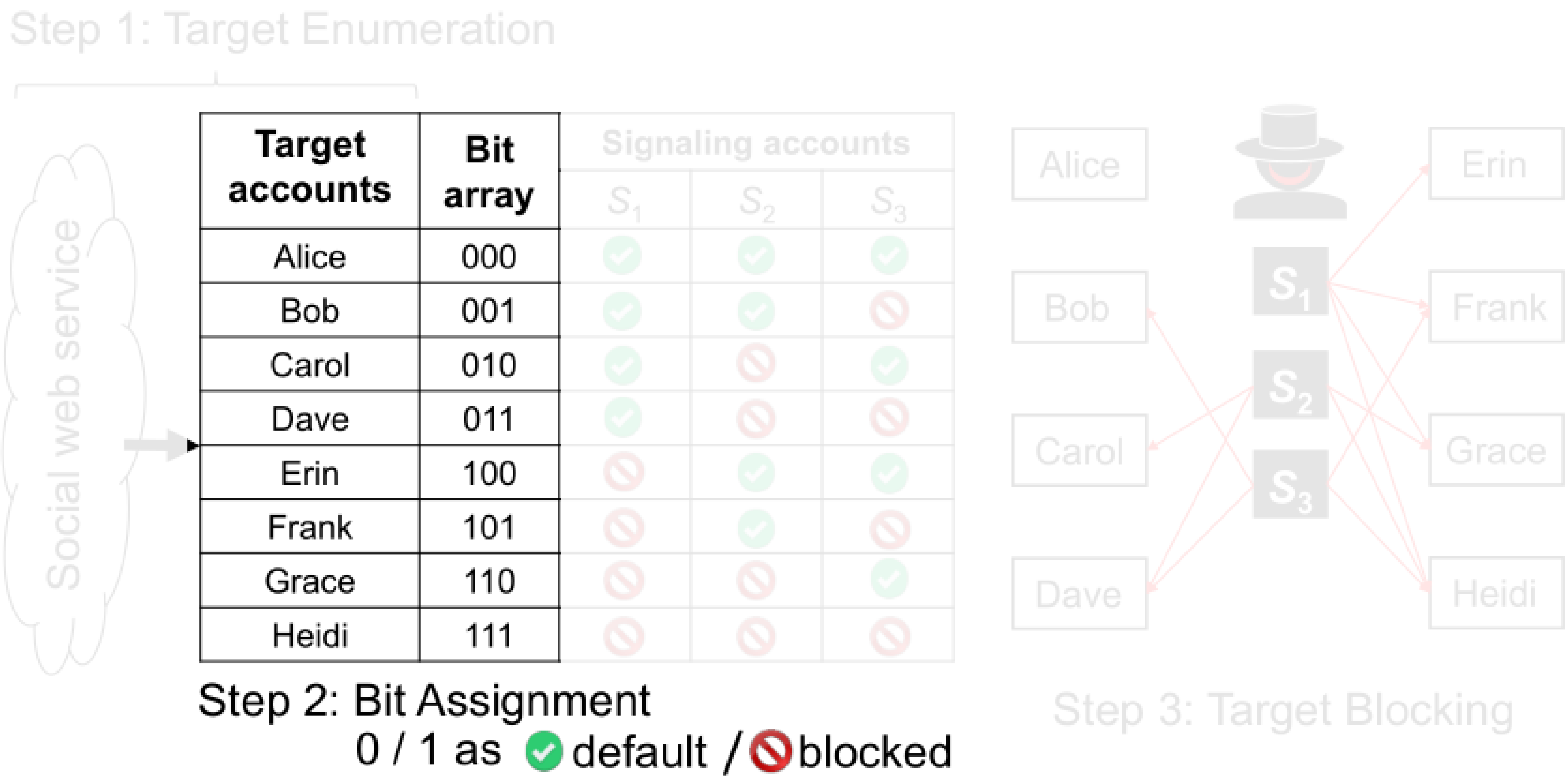
Target accounts	Bit array	Signaling accounts		
		S_1	S_2	S_3
Alice	000	✓	✓	✓
Bob	001	✓	✓	⊘
Carol	010	✓	⊘	✓
Dave	011	✓	⊘	⊘
Erin	100	⊘	✓	✓
Frank	101	⊘	✓	⊘
Grace	110	⊘	⊘	✓
Heidi	111	⊘	⊘	⊘

Step 2: Bit Assignment

0 / 1 as ✓ default / ⊘ blocked

Step 3: Target Blocking

Side-Channel Control Phase

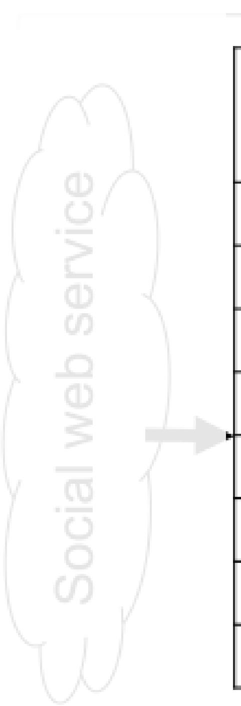


Target accounts	Bit array	Signaling accounts		
		S_1	S_2	S_3
Alice	000	✓	✓	✓
Bob	001	✓	✓	⊘
Carol	010	✓	⊘	✓
Dave	011	✓	⊘	⊘
Erin	100	⊘	✓	✓
Frank	101	⊘	✓	⊘
Grace	110	⊘	⊘	✓
Heidi	111	⊘	⊘	⊘

Step 2: Bit Assignment
0 / 1 as ✓ default / ⊘ blocked

Side-Channel Control Phase

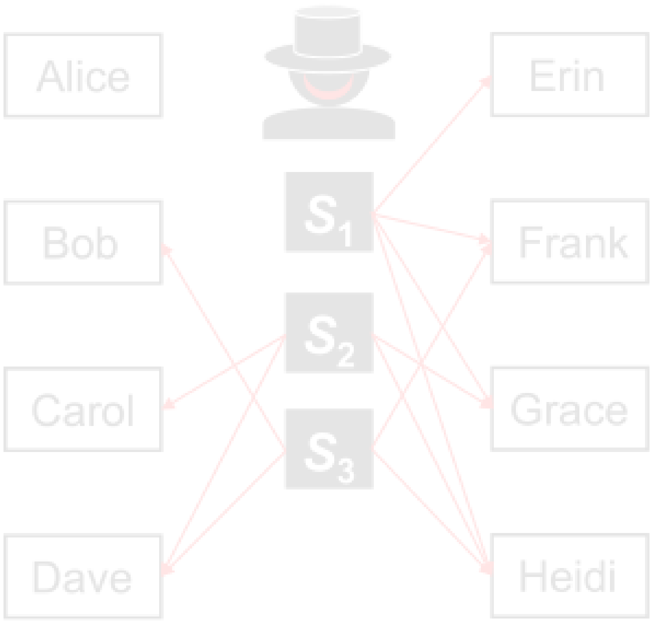
Step 1: Target Enumeration



Prepared by an attacker

Target accounts	Bit array	Signaling accounts		
		S_1	S_2	S_3
Alice	000	✓	✓	✓
Bob	001	✓	✓	🚫
Carol	010	✓	🚫	✓
Dave	011	✓	🚫	🚫
Erin	100	🚫	✓	✓
Frank	101	🚫	✓	🚫
Grace	110	🚫	🚫	✓
Heidi	111	🚫	🚫	🚫

Step 2: Bit Assignment
0 / 1 as ✓ default / 🚫 blocked

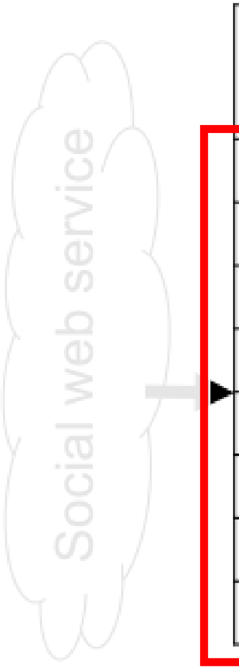


Step 3: Target Blocking

Side-Channel Control Phase

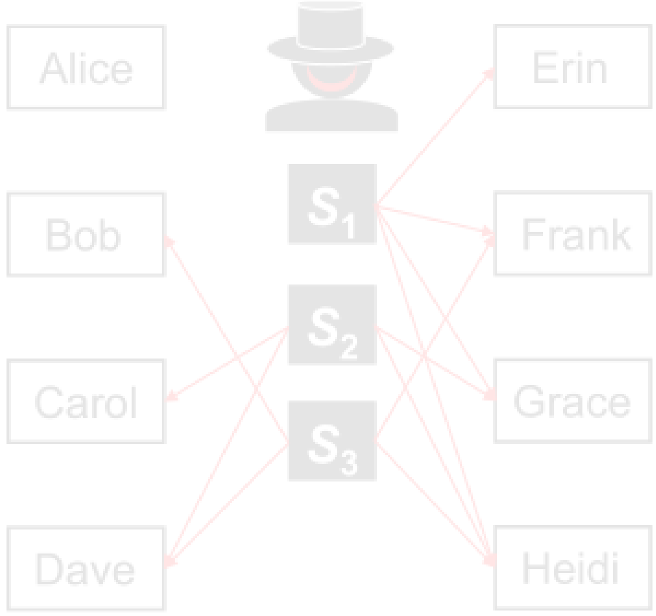
An attacker needs to prepare only **m** signaling accounts to cover **2^m** targets

Step 1: Target Enumeration



Target accounts	Bit array	Signaling accounts		
		S_1	S_2	S_3
Alice	000	✓	✓	✓
Bob	001	✓	✓	⊘
Carol	010	✓	⊘	✓
Dave	011	✓	⊘	⊘
Erin	100	⊘	✓	✓
Frank	101	⊘	✓	⊘
Grace	110	⊘	⊘	✓
Heidi	111	⊘	⊘	⊘

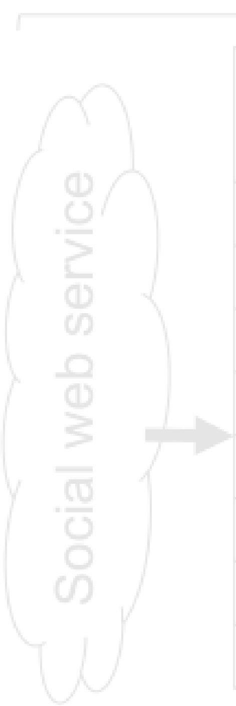
Step 2: Bit Assignment
0 / 1 as ✓ default / ⊘ blocked



Step 3: Target Blocking

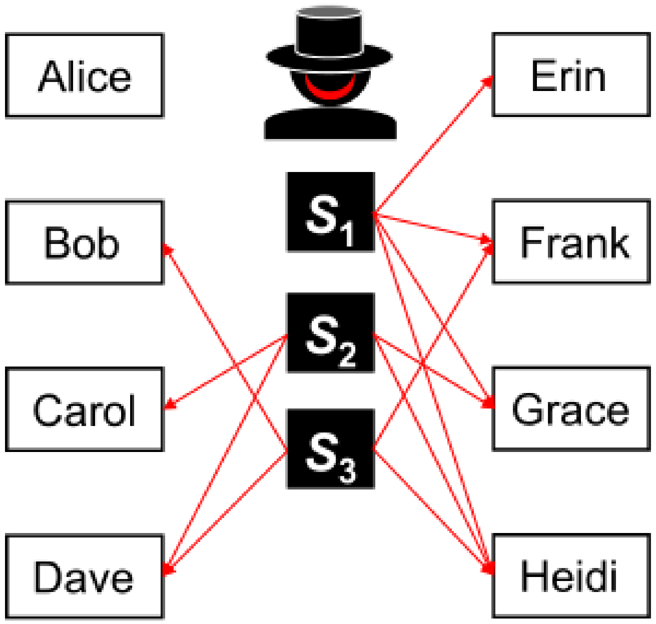
Side-Channel Control Phase

Step 1: Target Enumeration



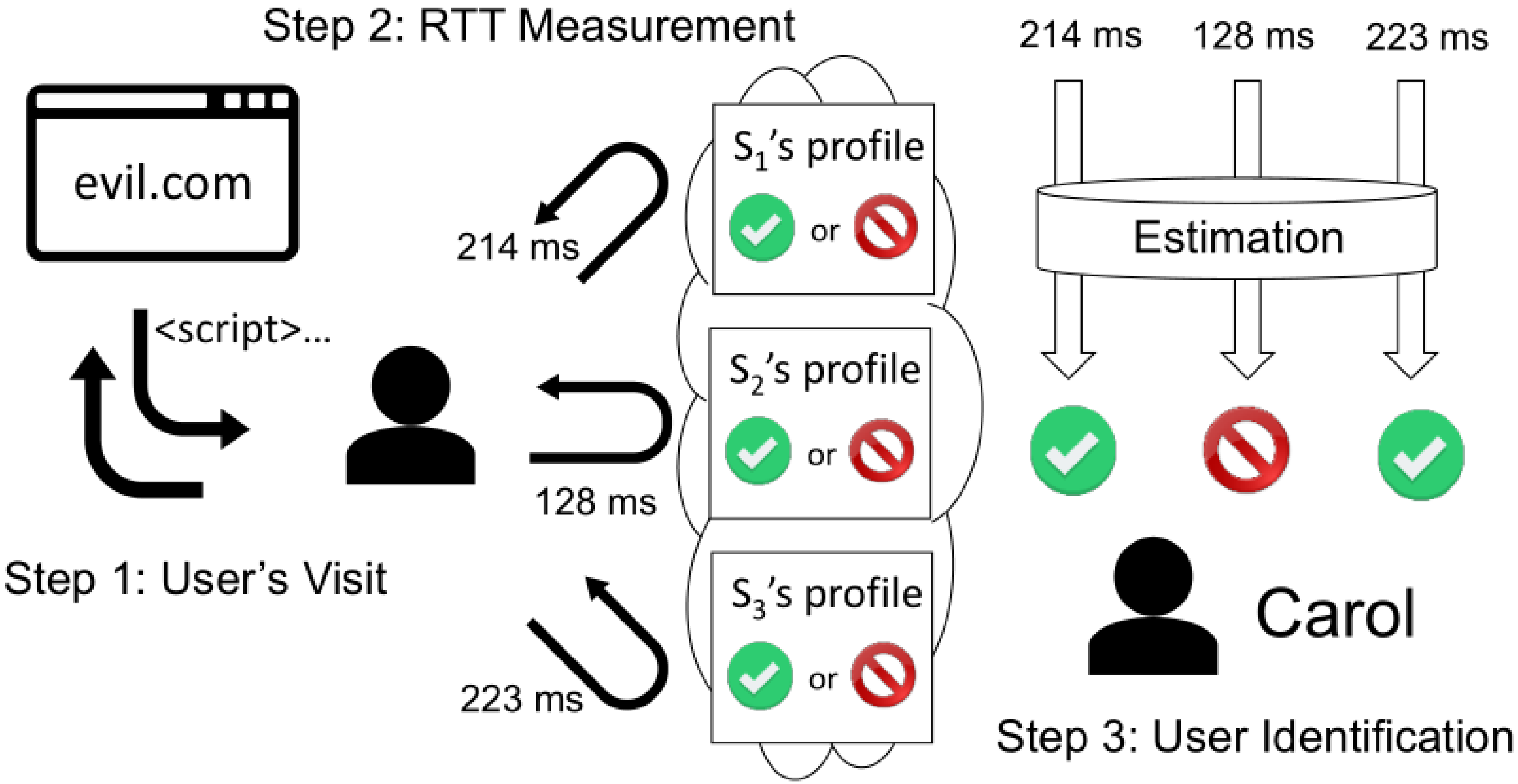
Target accounts	Bit array	Signaling accounts		
		S_1	S_2	S_3
Alice	000	✓	✓	✓
Bob	001	✓	✓	⊘
Carol	010	✓	⊘	✓
Dave	011	✓	⊘	⊘
Erin	100	⊘	✓	✓
Frank	101	⊘	✓	⊘
Grace	110	⊘	⊘	✓
Heidi	111	⊘	⊘	⊘

Step 2: Bit Assignment
0 / 1 as ✓ default / ⊘ blocked

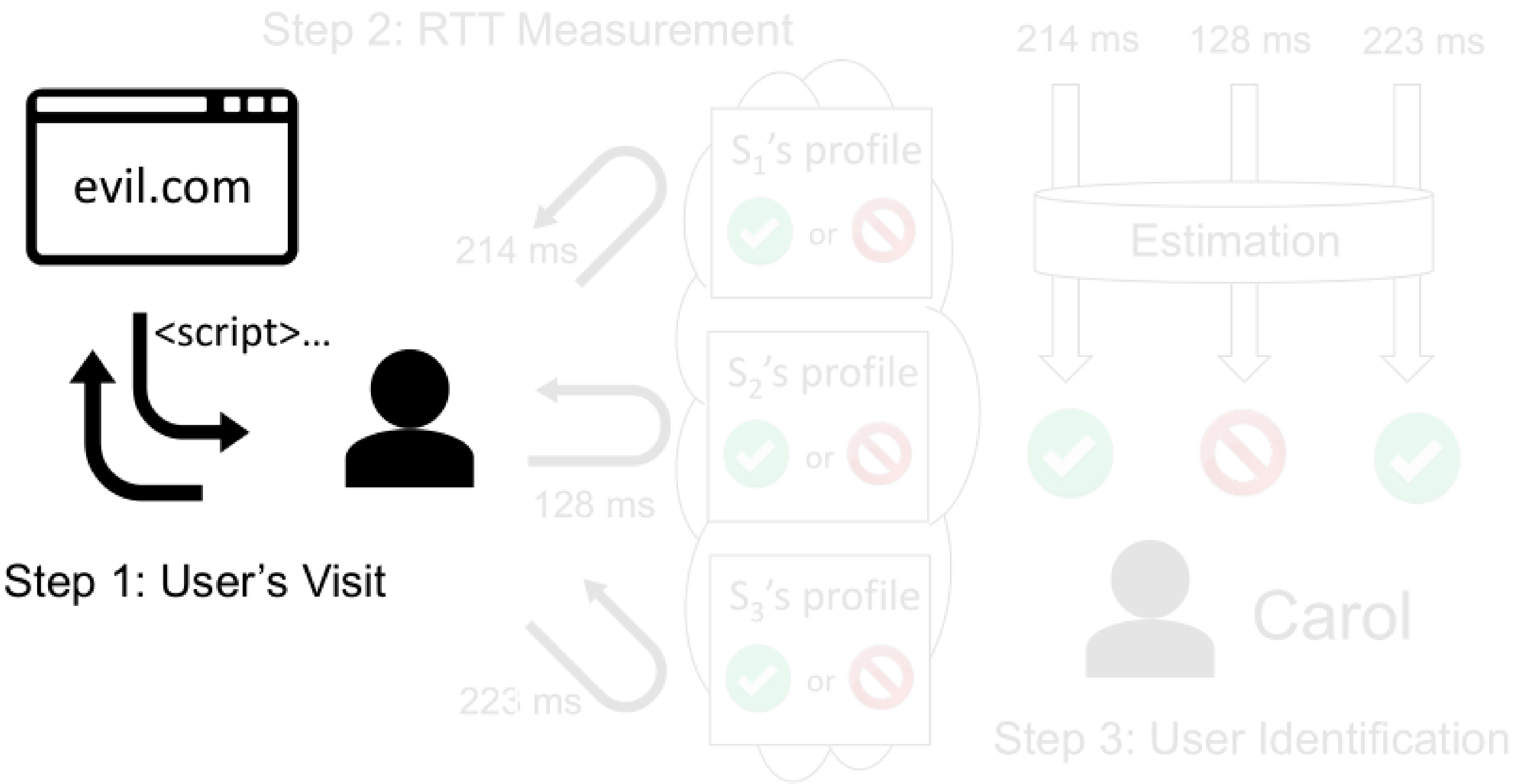


Step 3: Target Blocking

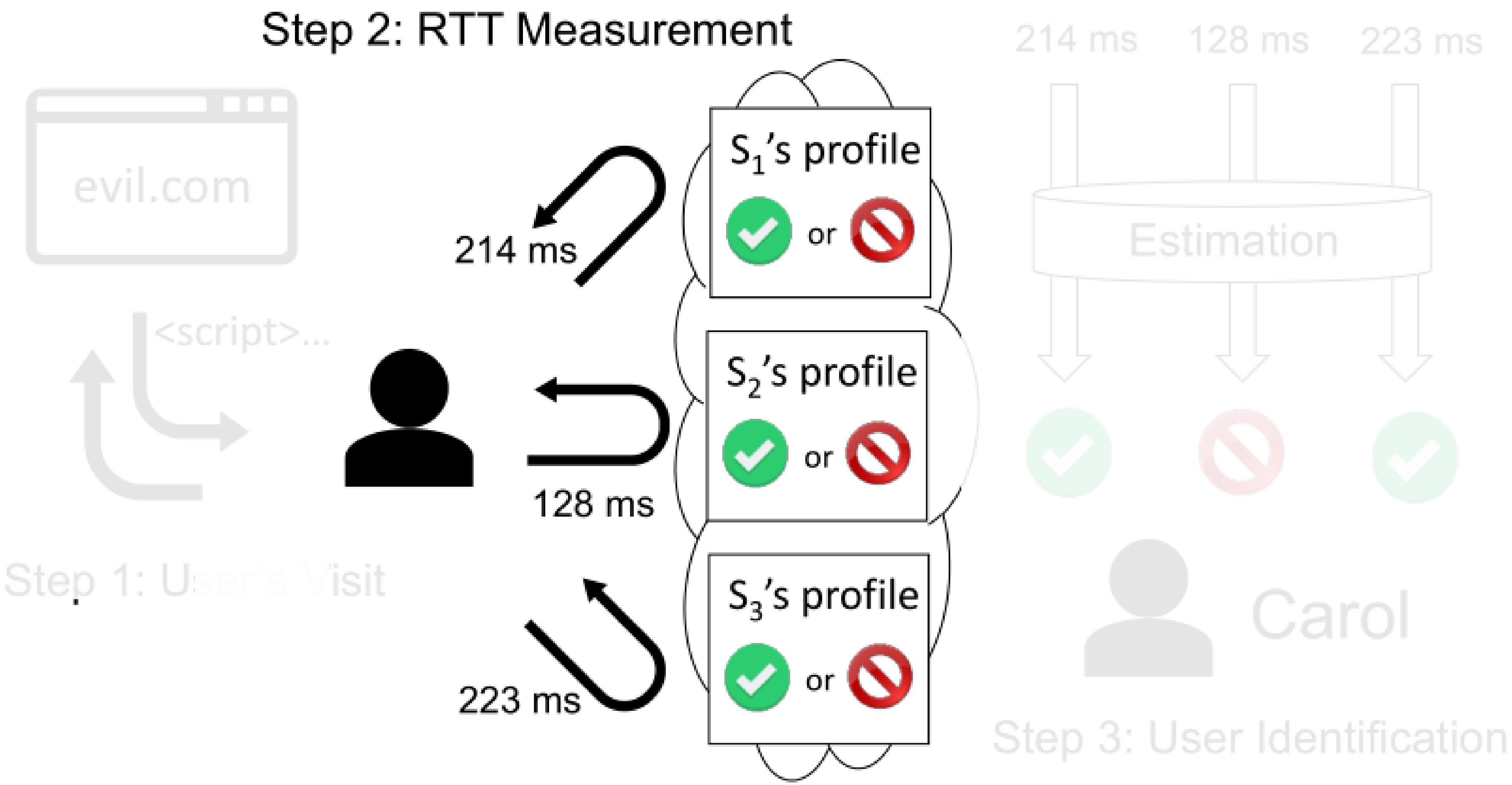
Side-Channel Retrieval Phase



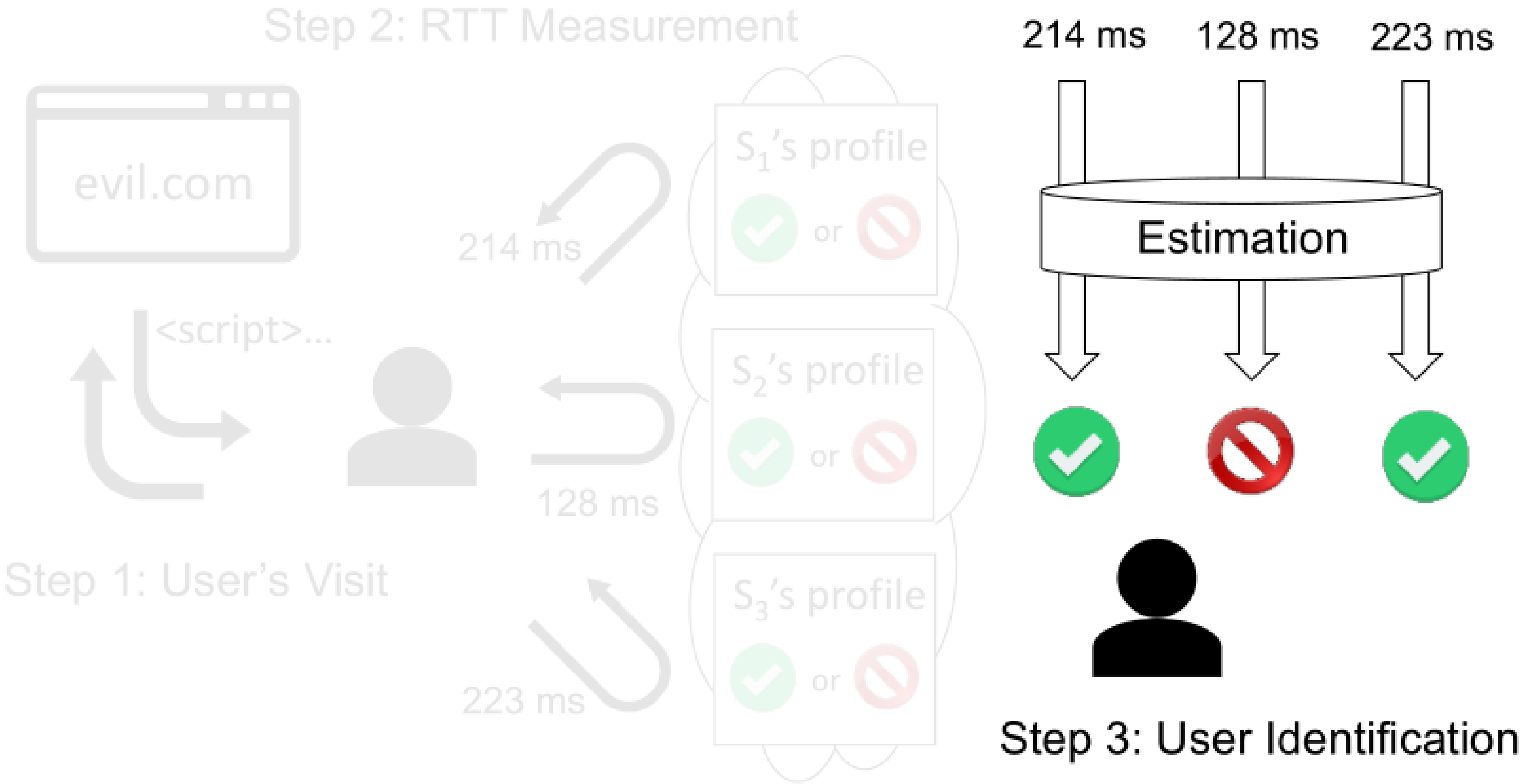
Side-Channel Retrieval Phase



Side-Channel Retrieval Phase

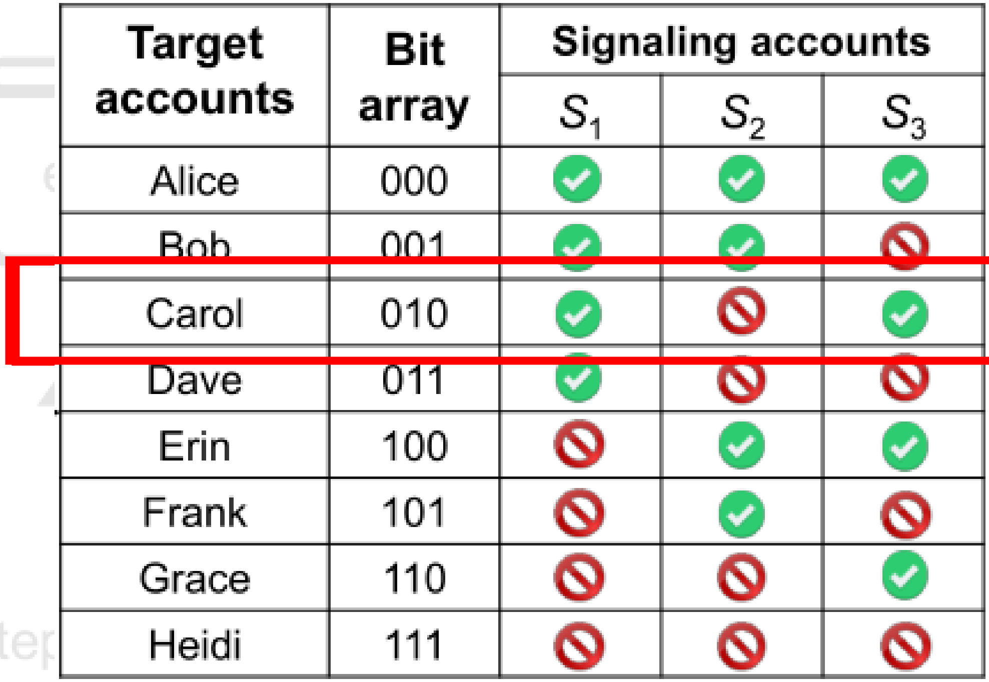


Side-Channel Retrieval Phase

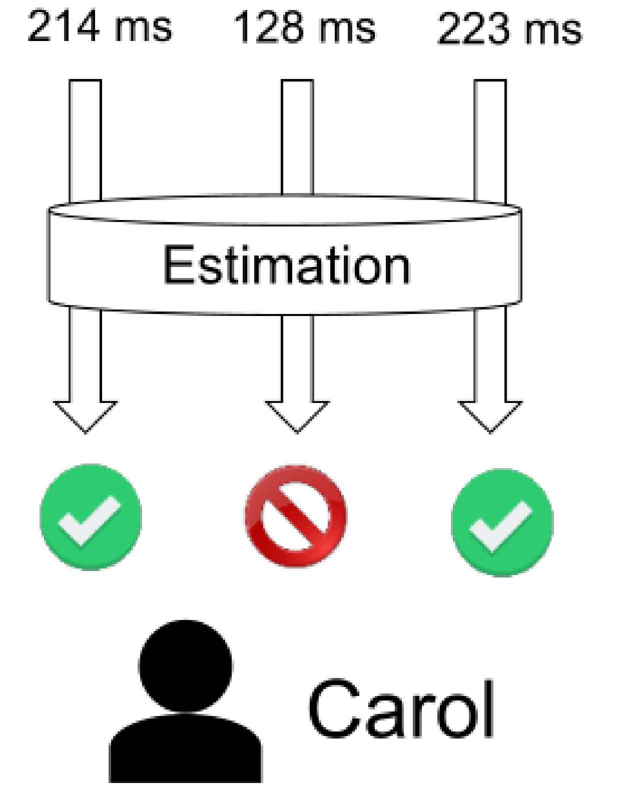


Side-Channel Retrieval Phase

Step 2: RTT Measurement



Target accounts	Bit array	Signaling accounts		
		S_1	S_2	S_3
Alice	000	✓	✓	✓
Bob	001	✓	✓	🚫
Carol	010	✓	🚫	✓
Dave	011	✓	🚫	🚫
Erin	100	🚫	✓	✓
Frank	101	🚫	✓	🚫
Grace	110	🚫	🚫	✓
Heidi	111	🚫	🚫	🚫



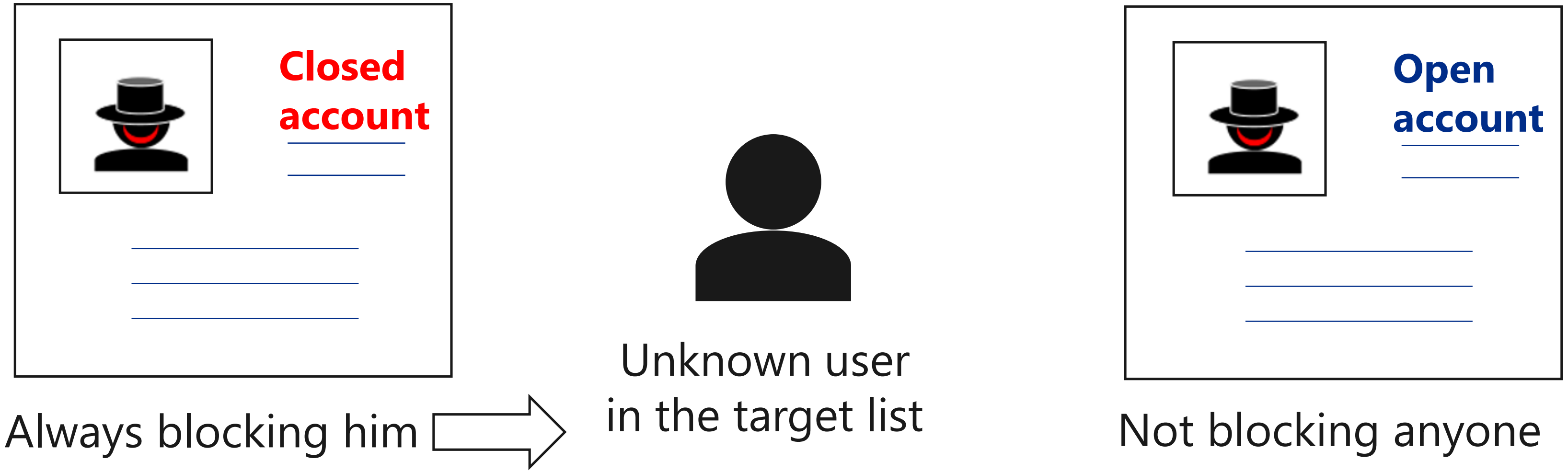
Step 3: User Identification

Aren't RTTs dependent on user environment?

- Our method prepares 2 extra accounts:

  Closed account blocks all users included in the list of targets

  Open account does not block any users at all



- It is useful to determine the threshold of RTT

Estimation Procedure

1. A website visitor is forced to send requests to closed/open accounts
   - Repeat 30 times for each account
   - Let C and O be the 5th-percentiles of the RTT values measured for the closed/open accounts, respectively

2. The visitor is forced to send requests to signaling accounts
   - Repeat **k** times for each account
   - Let R_j be the 5th-percentile of the RTT values measured for the j-th signaling account, S_j

3. The attacker estimates the visitor's status and retrieves bit array
   - The visitor is blocked by S_j if R_j is closer to C than O
   - The visitor is non-blocked by S_j if R_j is closer to O than C

Extensions

- Error-correction Coding
  - A few estimation errors can be corrected efficiently
  - We adopt the Reed-Solomon code in this work
    - ➢ Just add redundant bits for each target

- User-space Partitioning
  - The size of the target list of our attack can be constrained by the maximum blocks of the service.
  - The target list is enlarged by partitioning the user space and running an additional measurement stage.

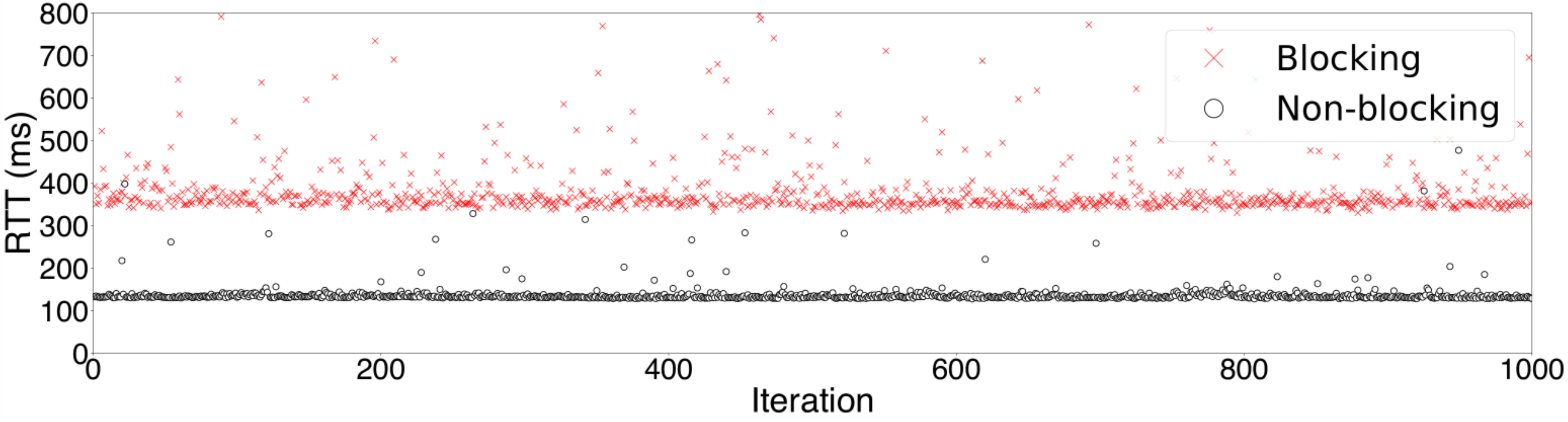
➡ Detailed in the whitepaper

Demo

Field Experiments

Distinguishability of RTTs

- The success of our attack depends on distinguishability of RTTs for blocking and non-blocking accounts



Distributions of RTTs for blocking and non-blocking in Facebook

Impact on the Real World

- We tested whether the RTTs for blocking/non-blocking accounts were statistically distinguishable in popular services
  - Applying Mann-Whitney U test
  - Distinguishable if *p*-value ≦ 0.01

- We found at least 12 popular services are vulnerable

SNS	Facebook, Twitter, Tumblr, Instagram, Google+, Medium
Auction	eBay
Game	Xbox Live, Roblox
Dating and Porn	PornHub, Xvideos, Ashley Madison

Accuracy of estimating a single bit

TBR: The rate of detecting the blocking user as a blocking

TNBR: The rate of detecting the non-blocking user as a non-blocking

	Facebook		**Twitter**		**Tumblr**	
k (# of trials)	TBR	TNBR	TBR	TNBR	TBR	TNBR
1	1.00	0.98	0.99	0.99	0.67	0.99
3	1.00	1.00	1.00	0.99	0.89	0.99
5	1.00	1.00	1.00	0.97	0.95	0.98
10	1.00	1.00	1.00	1.00	0.98	1.00
20	1.00	1.00	1.00	1.00	1.00	1.00
30	1.00	1.00	1.00	1.00	1.00	1.00

It was negligible to be affected by the PC performance and the browser type

Attack Success Rate in the Wild

- Use 20 real accounts as targets
  - In Facebook, Twitter, and Tumblr
- Assign random 24 bits for each account
  - Covering maximum 2^{24} targets
- Add redundant 8 bits for the Reed-Solomon code
  - With 4-bits block length, which enables it to collect one block error

# of accounts used for this experiment

	Target accounts	An attacker-controlled		Total
		signaling accounts	Redundant	
Facebook	20	24	8	52
Twitter	20	24	8	52
Tumblr	20	24	8	52

Attack Success Rate in the Wild (cont.)

- Use three different network environments
  - Wired LAN, Wi-Fi, and Tethering

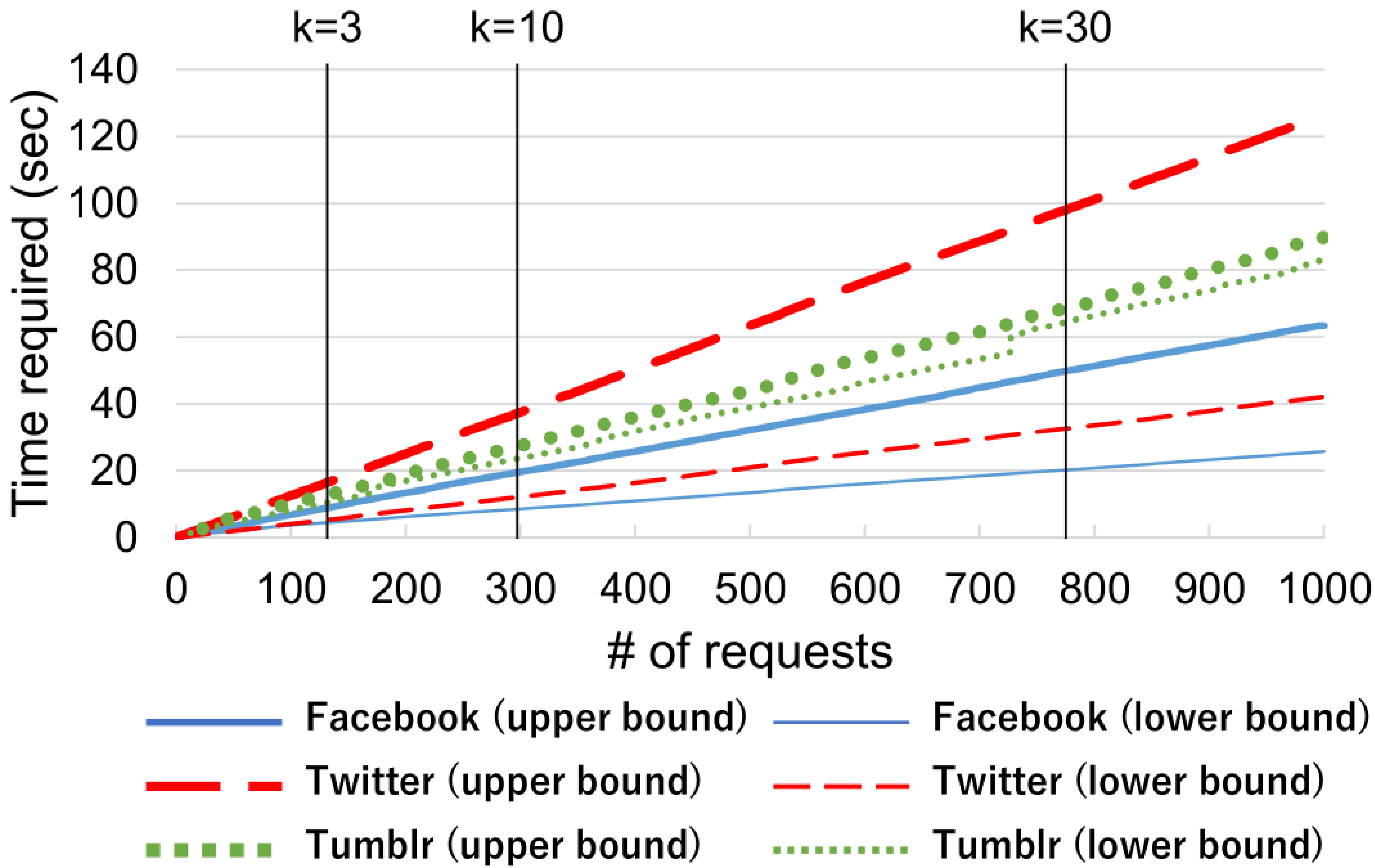
	Facebook/Wired	Twitter/Wi-Fi	Tumblr/Tethering
Success rate	0.95(19/20)	1.00(20/20)	1.00(20/20)
Success rate (with reed-solomon)	1.00(20/20)	1.00(20/20)	1.00(20/20)

Failure case

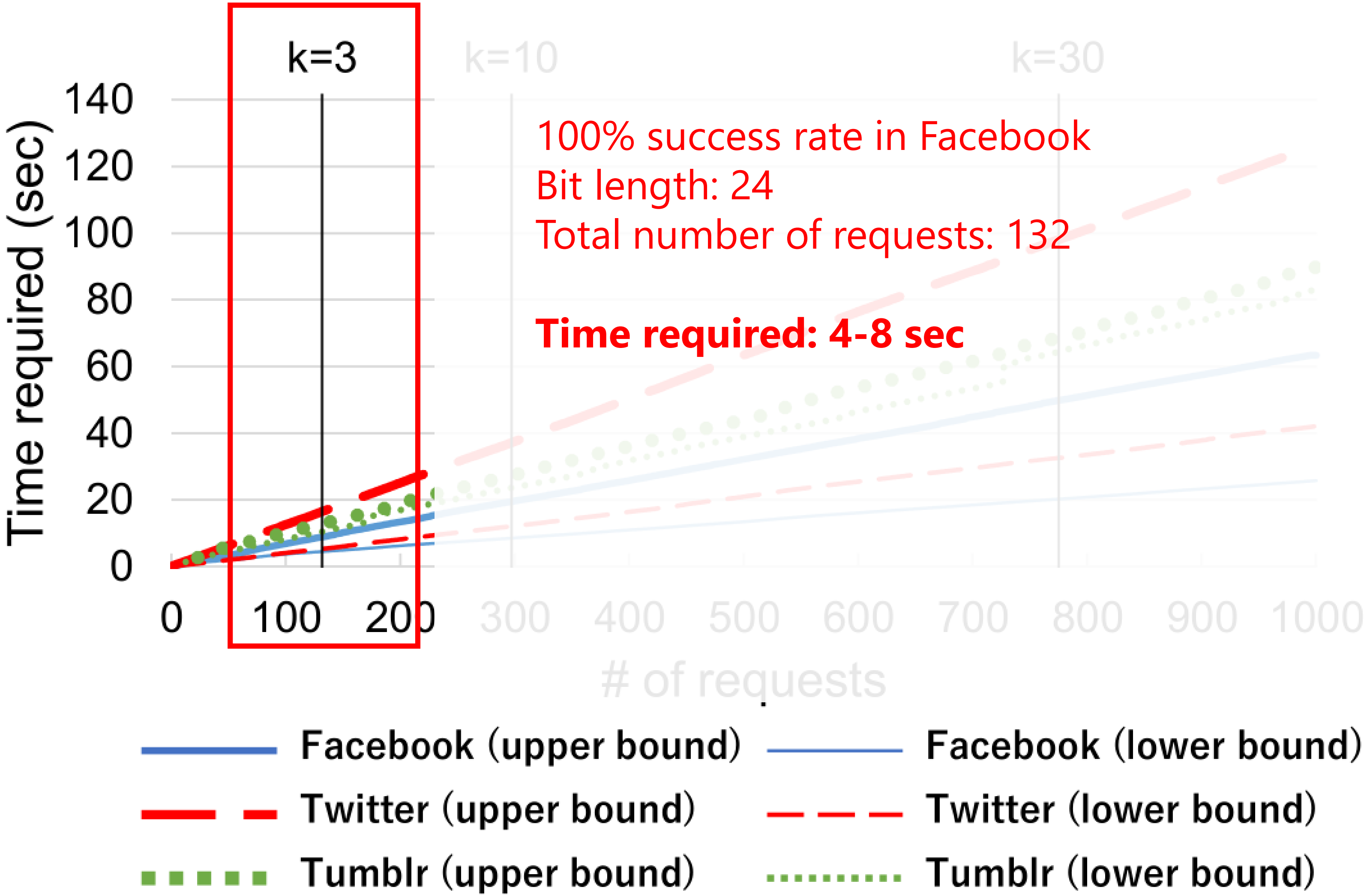
- 502 response are returned over 1 second
  - One bit error occurred, but it was corrected

Ultimately, user identification attack succeeded in all cases

Time to Complete the Attack



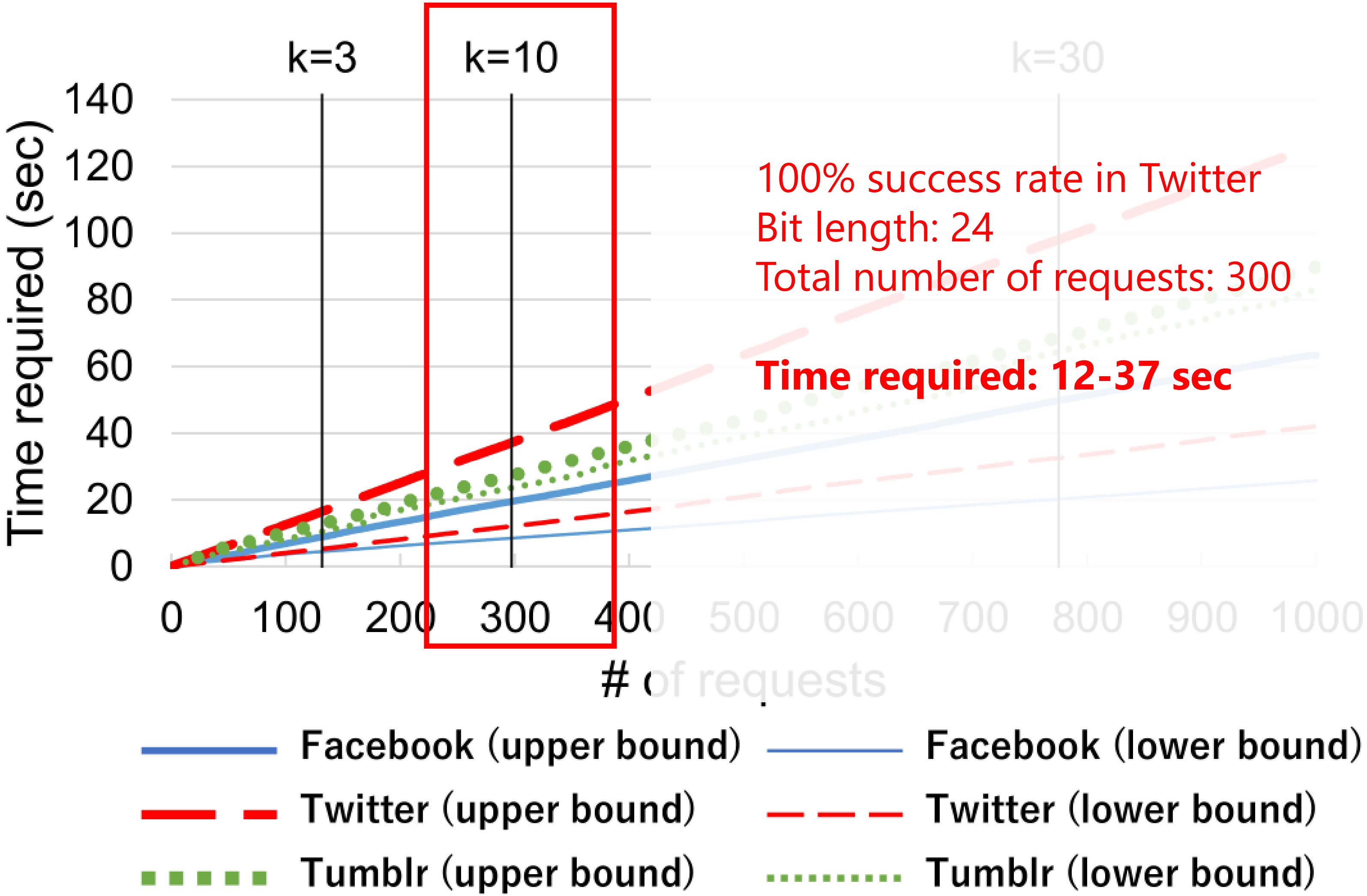
Time to Complete the Attack



100% success rate in Facebook
Bit length: 24
Total number of requests: 132

Time required: 4-8 sec

Time to Complete the Attack



100% success rate in Twitter
Bit length: 24
Total number of requests: 300

Time required: 12-37 sec

Discussions

Pioneer Work

G. Wondracek, T. Holz, E. Kirda, and C. Kruegel,

- *"A Practical Attack to De-anonymize Social Network Users"*

in IEEE S&P '10

  - has a similar goal
  - combines group membership information
  - depends on the "history stealing attack"

no longer feasible in the latest browsers
to the best of our knowledge

- Our work
  - leverages the *user blocking mechanism*
    - perfectly **attacker-controllable**
  - employs the cross-site timing attack
    - conventional, but even still available
  - demonstrates for the widespread type of web services
    - SNS, Shopping, Game, Dating, and Porn

Visibility Control in Social Webs

- Other feature whose visibility of a user is changed
  - Friendship
  - Membership of user group
  - Image sharing
  - …

	Blocking	Invitation	Subscribe
Attacker controllable	Yes	Yes	No
Notice to target	No	Yes	Yes
Require approval action	No	Depends	Yes

- User blocking tends not to have a limit (rate limit, upper limit)

Mobile Environment

- The RTTs can be identified even with the mobile browser

- Users of mobile platforms typically access social web services through dedicated mobile apps

- The mobile attack is established under some assumptions
  - Social plugin
  - Single Sign On
  - Webview

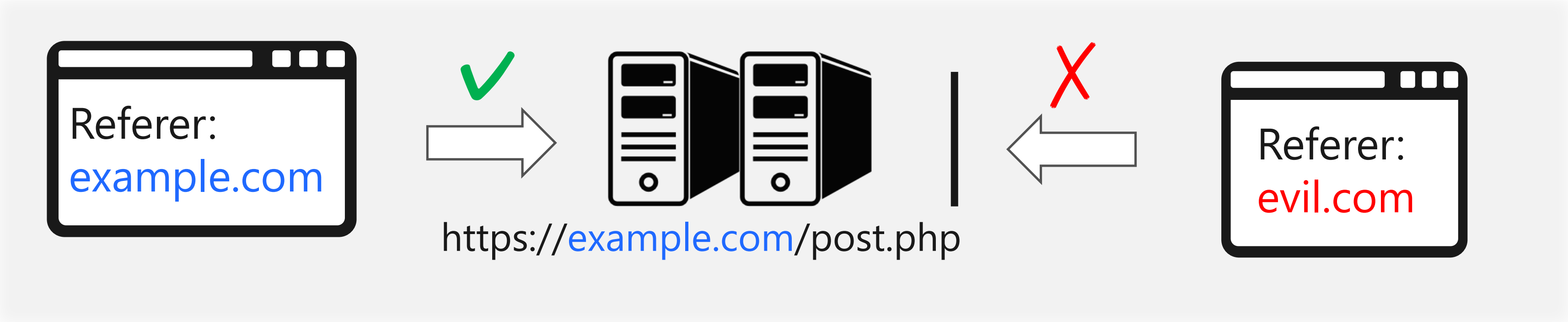
Defenses and Our Efforts

Possible Defenses

- Web Services
  - Same-site attribute
  - Place holder page
  - Intentional delay
- Browser vendors
  - Same-site attribute
  - Interrupting anomaly requests
  - Intentional delay
- Users
  - Secret mode
  - Sign out
  - NoScript

Typical CSRF defense

- Verify referer or CSRF token

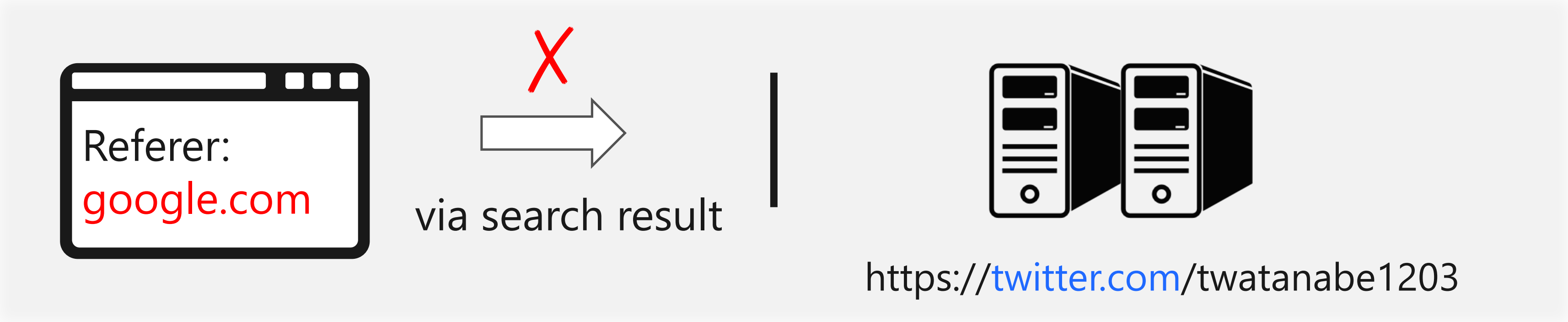


Referer: example.com ✓

https://example.com/post.php

Referer: evil.com ✗

- Concern: Profile pages are often accessed from other sites



Referer: google.com ✗

via search result

https://twitter.com/twatanabe1203

SameSite Attribute

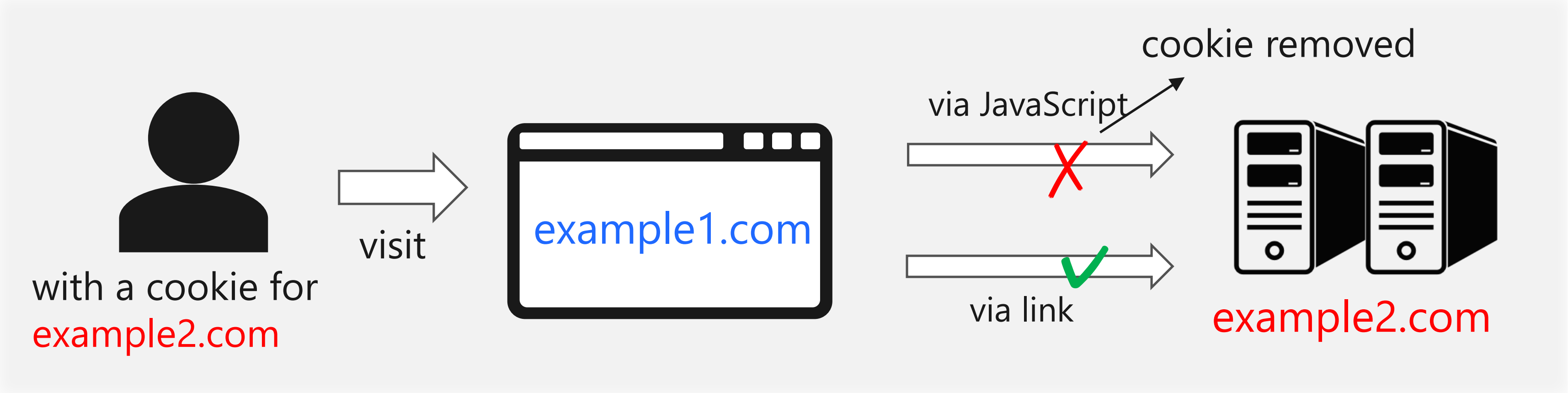
- An option proposed by google to prevent the browser from sending this cookie along with cross-site requests
- Usage:

```
Set-Cookie: sid=xxxx; path=/; samesite=lax
```

or

```
Set-Cookie: sid=xxxx; path=/; samesite=strict
```

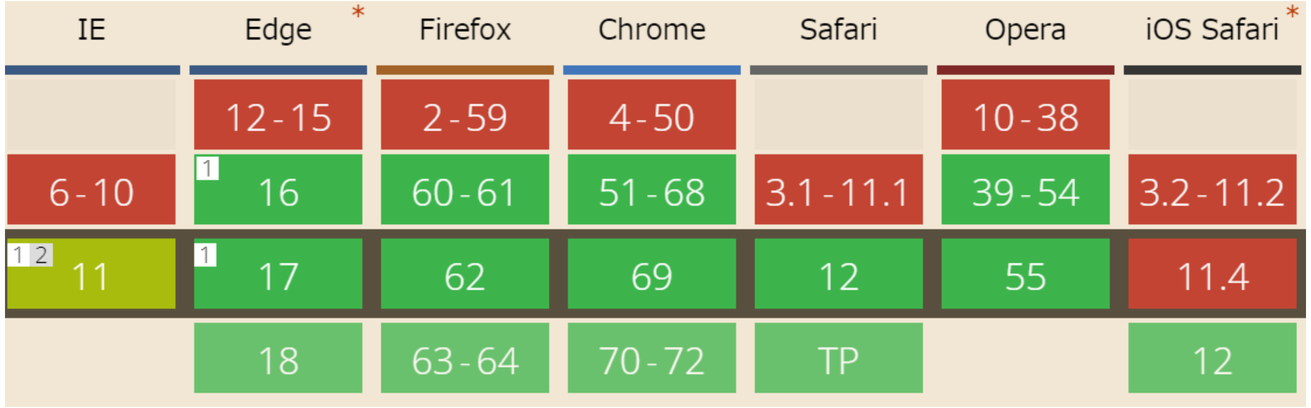
- Case of “samesite=lax”



- At first, browsers other than Chromium did not support the SameSite attribute.

Responsible Disclosure

- **Twitter** have adopted Same-site Cookies and Referer-based defense
  - The latter principle is similar to place holder page
- **Major browsers** have supported Same-site Cookies
  - The result of the request by us and Twitter



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *
	🟥 12-15	🟥 2-59	🟥 4-50		🟥 10-38	
🟥 6-10	🟩 [1] 16	🟩 60-61	🟩 51-68	🟥 3.1-11.1	🟩 39-54	🟥 3.2-11.2
🟨 [1] [2] 11	🟩 [1] 17	🟩 62	🟩 69	🟩 12	🟩 55	🟥 11.4
	🟩 18	🟩 63-64	🟩 70-72	🟩 TP		🟩 12

🟩 supported 🟥 unsupported

https://caniuse.com/#feat=same-site-cookie-attribute

- Several other services are also finished implementing defenses*

*We do not have permission to mention the brand names

Summary

- We presented a practical side-channel attack that identifies the social account of a website visitor
  - At least 12 services are vulnerable
  - It archives 100% success rate and takes as short as 4-8 sec

- It exploits the user-blocking mechanism, or *the visibility control property*, commonly available in most social web services today

- We have successfully addressed this attack by collaborative working with service providers and browser vendors.

Takeaways

- It should be noted that Internet users can be destroyed their anonymity by unexpected ways when using social web services.
- A feature that enables to control the visibility of other users like user blocking can introduce new information leakage paths to attackers.
- With all of the major browsers adopting the SameSite attribute, web developers obtained a robust means to prevent CSRF (including side-channel attacks).

Thank You!

Twitter: @twatanabe1203

E-mail: watanabe.takuya@lab.ntt.co.jp