

ブラウザフィンガープリンティングと TLSフィンガープリンティングによる ユーザトラッキングとその対策の研究

2020年2月15日 修士論文発表

大久保研究室 M2 中村綾花

1. 背景
2. フィンガープリンティング
 - 2.1 ブラウザフィンガープリンティング
 - 2.2 TLSフィンガープリンティング
 - 2.3 フィンガープリンティング対策
3. 先行研究
4. 既存のフィンガープリンティング
対策ツールの評価実験
 - 4.1 実験手法
 - 4.2 実験ツール
 - 4.3 実験結果
 - 4.4 考察
5. 対策手法の検討・実験
 - 5.1 対策の検討
 - 5.2 提案手法の実験
 - 5.3 実験構成
 - 5.4 実験結果
 - 5.5 考察
 - 5.6 結果考察
6. まとめ
7. 今後の課題

1. 背景

多くのWebサイトが何らかのユーザートラッキングを行っており、知らないうちに追跡されている

■ cookie

■ フィンガープリンティング

- 特にフィンガープリンティングは明示的に拒否しづらい

2014年のAcarらの研究

Alexaのアクセス数上位10,000サイト中5.5%
の

1. 背景

フィンガープリンティング対策ツールが存在

➤ 多くはブラウザ拡張機能で提供される



chrome ウェブストア

[ホーム](#) > [拡張機能](#) > Canvas Fingerprint Defender



Canvas Fingerprint Defender

提供元: Yubi

★★★★★ 21 | [仕事効率化](#) | ユーザー数: 17,721 人



Firefox Browser

ADD-ONS

[もっと見る](#)

[拡張機能](#)



ScriptSafe

作成者: [andryou](#)

Regain control of the web and surf more securely.



This is not a Recommended Extension. Make sure you trust it before installing.

1. 背景

2019年5月, Firefoxがデフォルトでフィンガープリント採取を防ぐオプションを実装

強化型トラッキング防止機能



トラッカーはあなたの習慣や興味に関する情報を収集するために、オンラインであなたを追跡します。Firefox はこれらのトラッカーや悪意のあるスクリプトの多くをブロックします。 [詳細](#)

[例外を管理...](#)

☐ 標準

保護と性能をバランスよく。ページが正しく機能するように読み込みます。

☐ 厳格

より強固な保護ですが、一部のサイトやコンテンツが機能しなくなる可能性があります。

 ソーシャルメディアトラッカー

 クロスサイトトラッキング Cookie

☒ トラッキングコンテンツ (すべてのウィンドウ)

 暗号通貨マイニング

 フィンガープリント採取

⚠ 注意！

トラッカーをブロックすると、一部のサイトの機能に影響がある可能性があります。すべてのコンテンツを読み込むには、トラッカーを許容してページを再読み込みします。 [詳細](#)

1.背景

ユーザのプライバシーの保護のため、
勝手にユーザトラッキングされないことが望ましい

➤フィンガープリンティング対策は重要技術

問題点を発見・改善することにより
より良い技術にできる

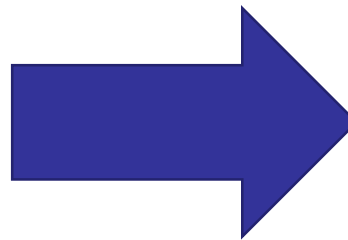
2. フィンガープリンティング

2. フィンガープリンティング

フィンガープリンティング(Fingerprinting)とは、
通信の特徴から人間の指紋(Fingerprint)のように
端末を識別する技術



Machine



Fingerprinting



Fingerprint
(識別情報)

2. フィンガープリンティング

フィンガープリンティングには2種類存在

1. アクティブフィンガープリンティング

サーバにアクセスしてきたクライアント上で情報を収集するプログラムを実行させ、クライアントを一意に識別する技術

- ブラウザフィンガープリンティング

2. パッシブフィンガープリンティング

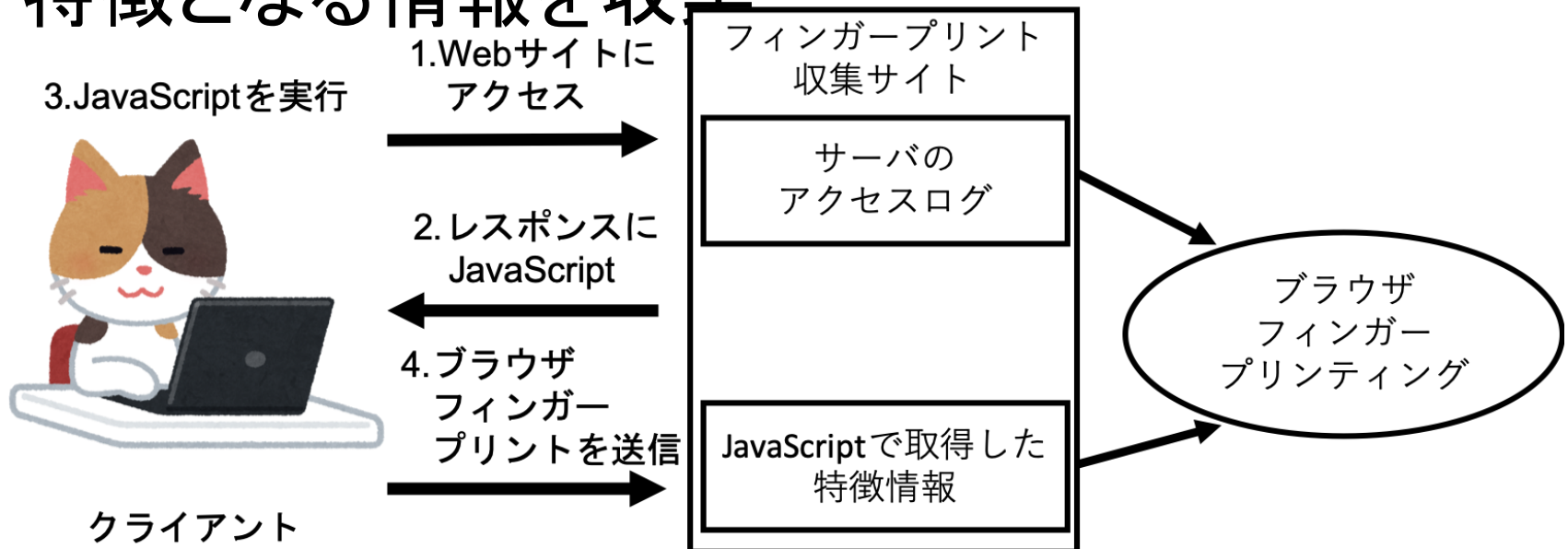
サーバにアクセスしてきたクライアントを、受信パケットの特徴を用いて識別する技術

- TCPフィンガープリンティング → OSの種類まで
- TLSフィンガープリンティング → ブラウザの種類まで

2.1 ブラウザフィンガープリンティング

クライアントのブラウザ上でJavaScriptを動作させ、

特徴となる情報を収集



2.1 ブラウザフィンガープリンティング

使用しているブラウザ特有の情報を収集して識別する

- User-Agent
- Accept
- 解像度
- タイムゾーン
- etc...
- プラグイン
- システムフォント
- Canvas API
- WebGL API

Fingerprint.js PRO pricing calculator

Fraud detection API demo

99.5% identification accuracy

Your ID:
WLmf0CUZU7hzPw0qyGmG

YOUR CURRENT VISIT

Location:	Setagaya-ku, Tokyo, Japan, 154-0017 (35.6468, 139.6546)
Browser:	Chrome 78.0.3904 on Mac OS X (10.15.1)
Incognito?	NO (open this demo in private tab to see it in action)
Bot?	NO
IP:	[REDACTED]



<https://fingerprintjs.com/demo>

2.2 TLSフィンガープリンティング

HTTPS通信を行う際に発生するネゴシエーションパケットからクライアントの特徴を収集する

■ ClientHello

- TLS Version
- Cipher Suites
- Extensions

ClientHelloの特徴は、ブラウザのTLS実装に大きく依存する

2.3 フィンガープリンティング対策

フィンガープリンティングを検知して防止、
もしくはフィンガープリントを偽装する

■ブラウザの拡張機能

- JavaScriptの実行防止
- 一部の情報の偽装



ホーム > 拡張機能 > Canvas Fingerprint Defender



Canvas Fingerprint Defender

提供元: Yubi

★★★★★ 21 | 仕事効率化 | ユーザー数: 17,72

■ブラウザ固有の機能

- Firefoxのフィンガープリント防止機能

3. 先行研究

■ ブラウザフィンガープリンティングに関する研究

Eckersleyの研究(2010)

ブラウザフィンガープリンティングで94.2%のクライアントのブラウザを一意に識別した

Vastelらの研究 (2018)

ブラウザ拡張によるフィンガープリンティング対策機能を利用するユーザについて、ブラウザの実装に依存するエラーや関数の戻り値を比較することで偽装されたフィンガープリントを検出できる

3. 先行研究

■ TLSフィンガープリンティングに関する研究

Husákらの研究

TLSのCipher SuitesとブラウザのUser-Agentで辞書を作成, 99.6%のクライアントのブラウザのUser-Agentを識別可能

Frolovらの研究

11 億個以上のTLS ClientHelloを収集・分析. Torなど通信の匿名化ツールにおける通信と通常のブラウザの通信と比較し, 検出可能であることを示した.

対策として, Go用のライブラリuTLSを作成.

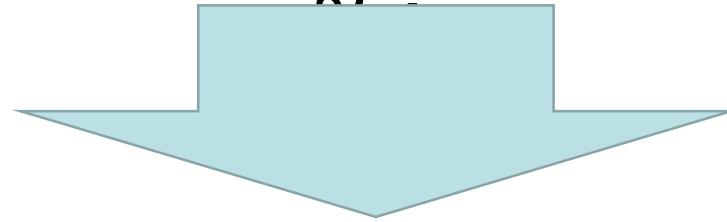
4. 既存のフィンガープリンティング対策ツールの評価実験

4. 既存のフィンガープリンティング 対策ツールの評価実験



情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

既存のフィンガープリンティング対策ツールは
ブラウザフィンガープリンティングに有効？
TLSフィンガープリンティングにも効果を発揮す
る？

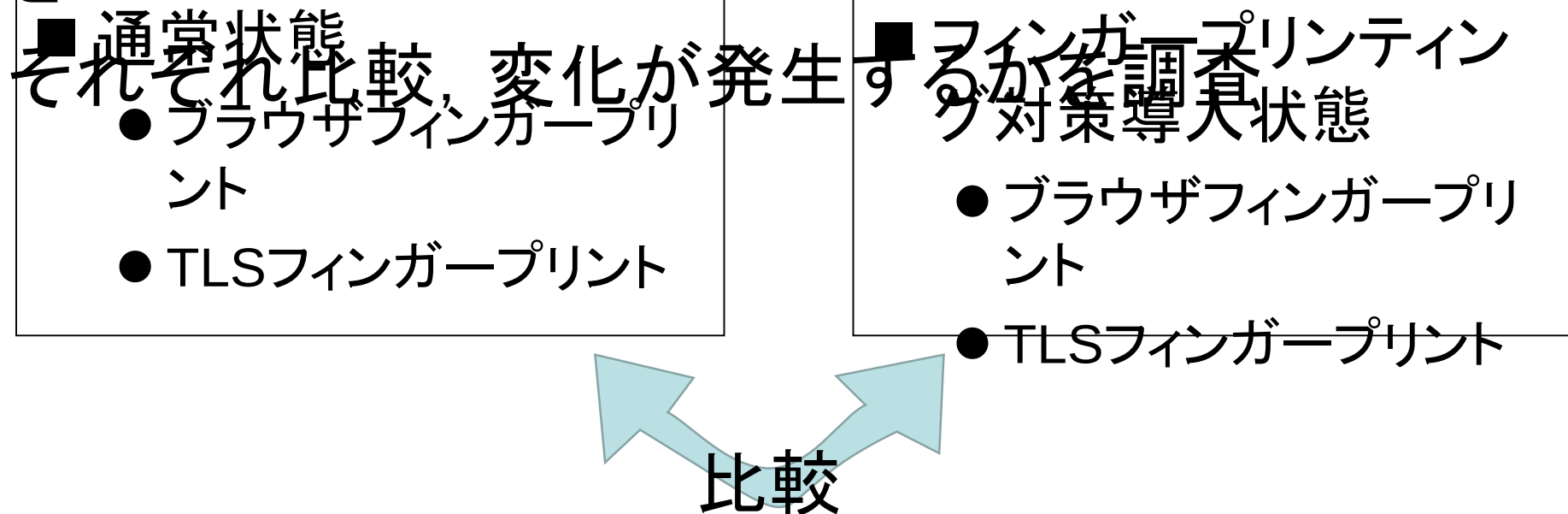


既存の人気フィンガープリンティング対策ツール
を調査

4.1 実験手法

既存のフィンガープリンティング対策ツールの性能評価

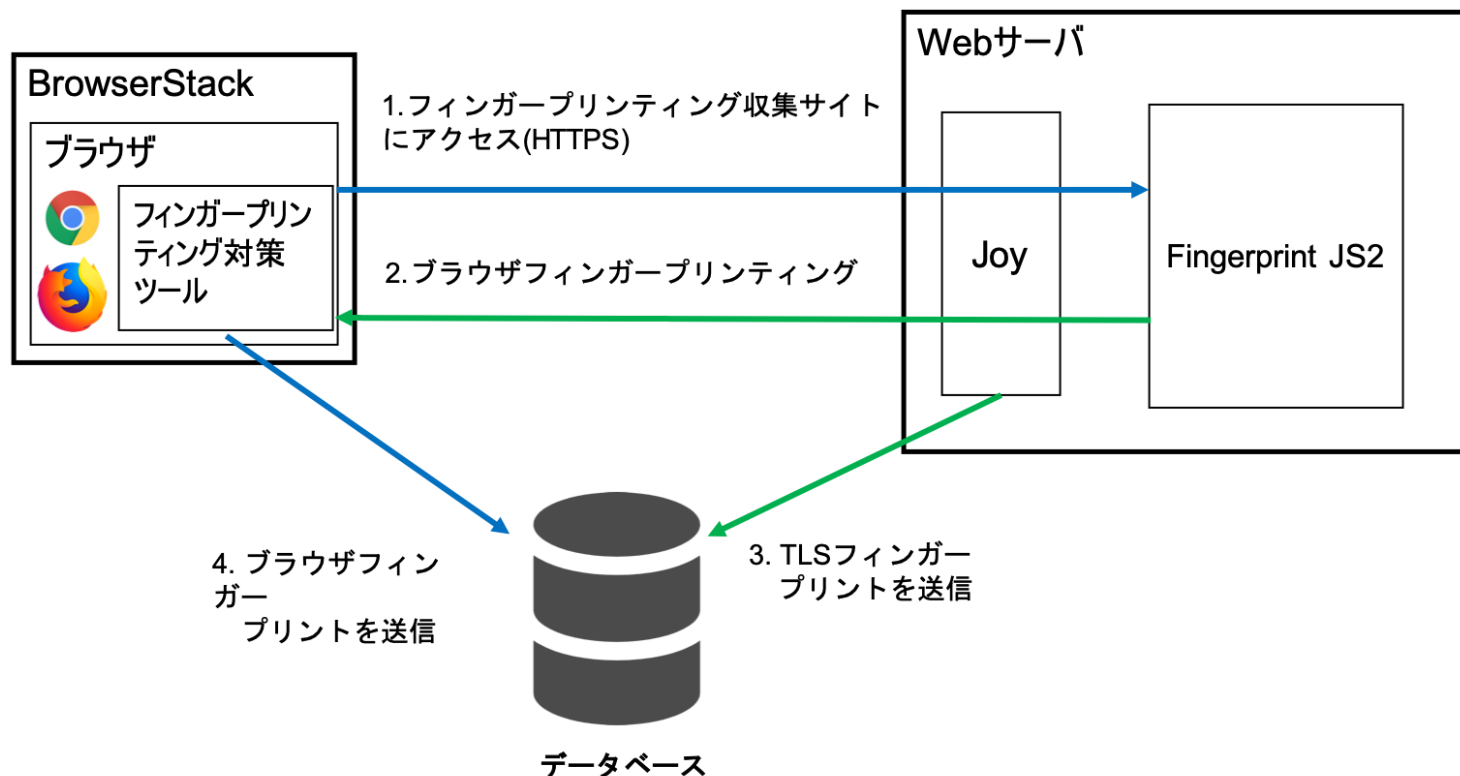
ブラウザの状態ごとに2種類のフィンガープリントを



4.2 実験手法

■ BrowserStackによるクリーンな仮想ブラウザを利用

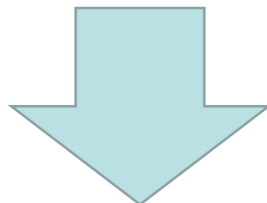
- Firefox
- Google Chrome



4.3 実験ツール

選定条件

- ブラウザ固有のもの
- 利用者が1000人以上のブラウザ拡張機能



Firefox, Google Chrome合わせて24個該当

4.3 実験ツール

ブラウザ	ツール名	機能概要
Firefox	トラッキング防止機能	ブラウザにデフォルトで存在するトラッキング防止機能
	Canvas Blocker	JavaScript で取得される情報を偽装またはブロック
	Canvas Defender	Canvas の偽装
	ScriptSafe	フィンガープリンティングのブロック
	Canvas Fingerprint Defender	Canvas の偽装
	Spoof Timezone	タイムゾーンの偽装
	AntiBrowserSpy TrackingBlocker	フィンガープリンティングスクリプトのブロック, User-Agent の偽装
	Trace	複数のトラッキングスクリプトからの保護
	AudioContext Fingerprint Defender	AudioContext の偽装
	WebGL Fingerprint Defender	WebGL の偽装

ブラウザ	ツール名	機能概要
Chrome	Canvas Fingerprint Defender	Canvas の偽装
	WebGL Fingerprint Defender	WebGL の偽装
	CanvasFingerprintBlock	Canvas のブロック
	AudioContext Fingerprint Defender	AudioContext の偽装
	Browser Plugs Fingerprint Privacy Firewall	複数のフィンガープリントの偽装
	Canvas Defender	Canvas の偽装
	ScriptSafe	フィンガープリンティングのブロック
	Canvas Blocker (Fingerprint protect)	Canvas の偽装
	JustBlock Security	広告のブロック, トラッキングスクリプトからの保護
	Random User-Agent	User-Agent の偽装
	Trace - Online Tracking Protection	複数のトラッキングスクリプトからの保護
	AntiBrowserSpy - TrackingBlocker	フィンガープリンティングスクリプトのブロック, User-Agent の偽装
	TunnelBear Blocker	広告と複数のトラッキングスクリプトのブロック
	Spoof Timezone	タイムゾーンの偽装

フィンガープリンティング 対策ツールの評価実験 実験結果

4.4 実験結果

ブラウザ	ツール名	ブラウザフィンガープリント	TLSフィンガープリント
Firefox	トラッキング防止機能	x	x
	Canvas Blocker	○	x
	Canvas Defender	○	x
	ScriptSafe	取得不可	x
	Canvas Fingerprint Defender	○	x
	Spoof Timezone	○	x
	AntiBrowserSpy TrackingBlocker	取得不可	x
	Trace	○	x
	AudioContext Fingerprint Defender	○	x
	WebGL Fingerprint Defender	○	x

変化: ○, 変化なし: x

ブラウザ	ツール名	ブラウザフィンガープリント	TLSフィンガープリント
Chrome	Canvas Fingerprint Defender	○	X
	WebGL Fingerprint Defender	○	X
	CanvasFingerprintBlock	○	X
	AudioContext Fingerprint Defender	○	X
	Browser Plugs Fingerprint Privacy Firewall	○	X
	Canvas Defender	X	X
	ScriptSafe	取得不可	X
	Canvas Blocker (Fingerprint protect)	○	X
	JustBlock Security	○	X
	Random User-Agent	○	X
	Trace - Online Tracking Protection	○	X
	AntiBrowserSpy - TrackingBlocker	取得不可	X
	TunnelBear Blocker	取得不可	X
	Spoof Timezone	○	X

4.5 考察

1. ChromeのCanvasDefender拡張機能
フィンガープリントの変化が起こらない
→ 対策効果が発揮されていない
 - 調査に利用したBrowserStackの問題
 - 環境上のLocalStorageにアクセスできないことによる
JavaScriptのエラー

4.5 考察

2. Firefoxのトラッキング防止機能

フィンガープリントの変化が起こらない

→ 対策効果が発揮されていない

➤ Disconnect社の提供する
ブラックリスト

➤ ユーザ側で編集が不可能



4.6 結果考察

24個のツール中，初期状態と比較した結果

■ブラウザフィンガープリント ■TLSフィンガープリント

- 変化: 17個
- 取得不可: 5個

- 全て変化なし

既存のフィンガープリンティング対策ツールは
TLSフィンガープリンティングに対して効果を発揮していない

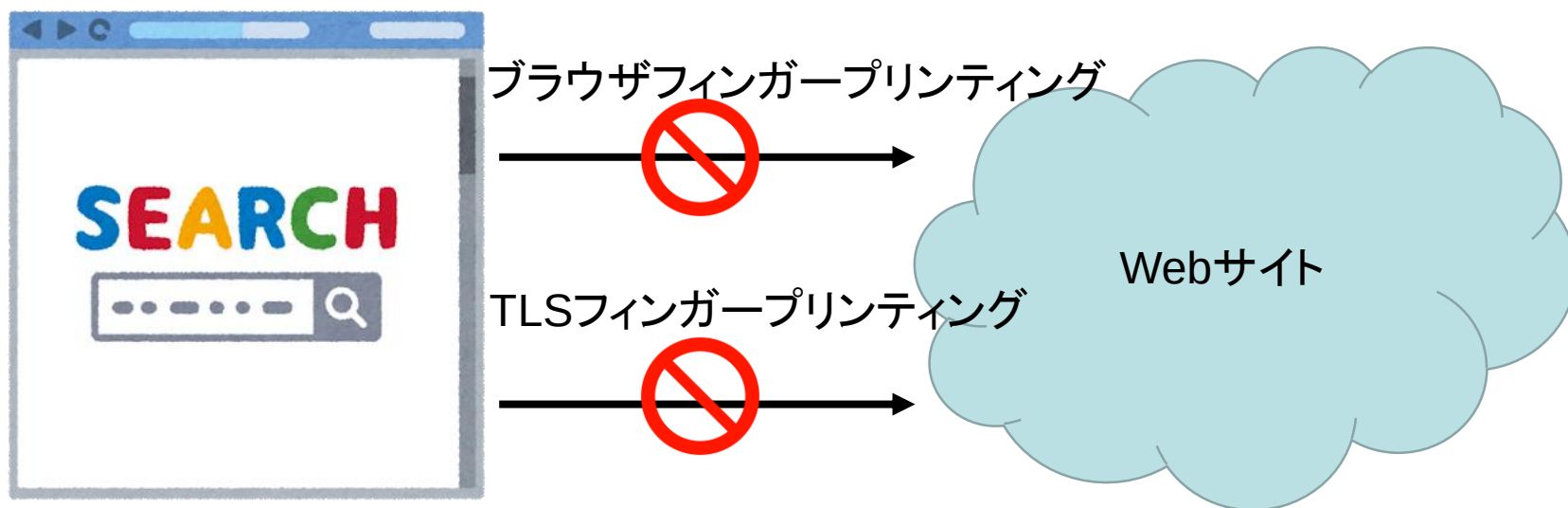


突合すれば検出されるおそれ
特徴として追跡される可能性

5. 対策手法の検討・実験

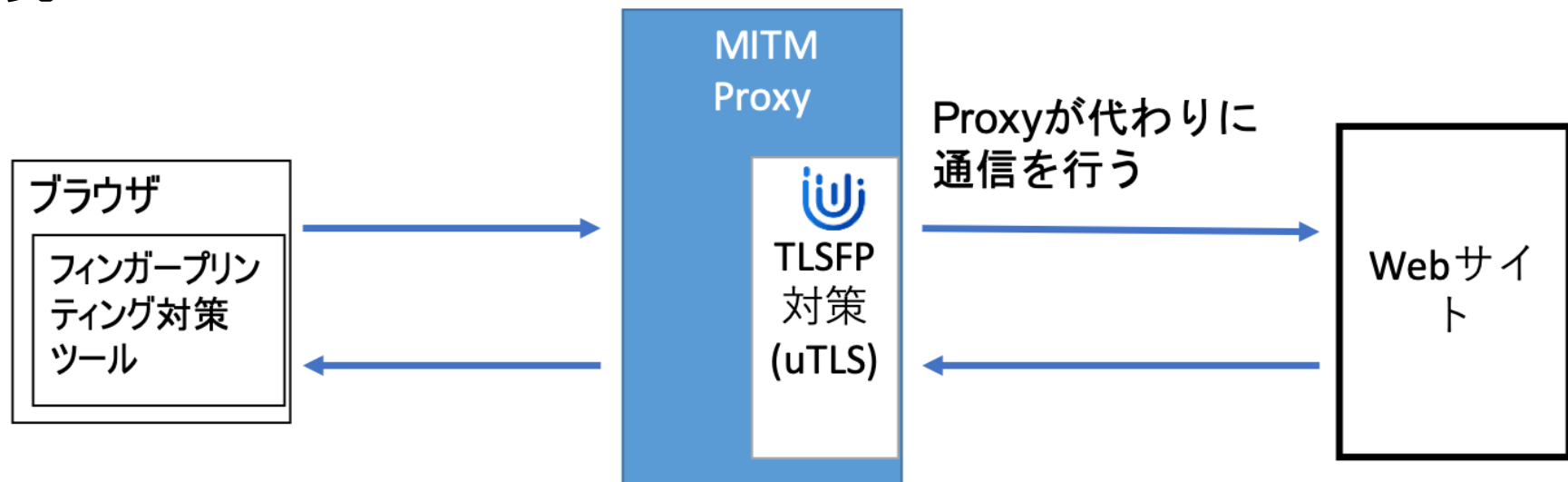
5.1 対策の検討

ブラウザフィンガープリンティング対策と同時に
TLSフィンガープリンティング対策を行えば良い



5.1 対策の検討

本研究では、
従来のブラウザフィンガープリンティング対策ツールを適用
Man In The Middleを行うProxyを設置、
TLSのClientHelloを変更して代わりに通信させることで実
現



5.2 提案手法の実験

1. MITM Proxy を構築, Proxy の証明書を実験端末に信頼できるものとしてインストール
2. 何も対策を施さない状態(通常状態), ブラウザフィンガープリンティング対策ツールを適用した状態(BFP対策), TLSフィンガープリンティング対策を適用した状態(TLSFP対策), ブラウザ・TLS両方のフィンガープリンティング対策を適用した状態(提案手法), それぞれの状態ですべて10回フィンガープリンティング収集サイトにアクセス
3. フィンガープリントが収集された回数と一意なフィンガープリントが生成された回数を比較

5.3 実験構成

■ 端末:

Windows10, macOS 各1台

■ ブラウザ:

Firefox, Google Chrome

■ ブラウザフィンガープリンティング対策ツール:

Canvas Fingerprint Defender
(Firefox, Chrome)

■ MITM Proxyの構築:

既存のプログラムを改変, Frolovらの
uTLSライブラリ(ClientHelloをランダムマイズ)
を実装

提案手法の実験 実験結果

5.4 実験結果

OS, ブラウザ	対策状況	ブラウザフィンガープリント		TLSフィンガープリント	
		Unique FP	総回数	Unique FP	総回数
Windows10 Firefox	通常状態	1	10	2	20
	BFP対策	10	10	2	20
	TLSFP対策	1	10	20	20
	提案手法	10	10	20	20
Windows10 Chrome	通常状態	1	10	1	20
	BFP対策	10	10	1	20
	TLSFP対策	1	10	20	20
	提案手法	10	10	20	20

5.4 実験結果

OS, ブラウザ	対策状況	ブラウザフィンガープリント		TLSフィンガープリント	
		Unique FP	総回数	Unique FP	総回数
macOS Firefox	通常状態	1	10	2	20
	BFP対策	10	10	2	20
	TLSFP対策	1	10	20	20
	提案手法	10	10	20	20
macOS Chrome	通常状態	1	10	1	20
	BFP対策	10	10	1	20
	TLSFP対策	1	10	20	20
	提案手法	10	10	20	20

5.5 考察

通常状態のFirefoxが2種類のTLSフィンガープリントを生成した事象

送信されるパケット中のextensionのパディングの有無により2種類生成

パディング
なし

```
(0303)(130113031302c02bc02fcca9cca8c02cc030c00ac009c013c01400330039002f0035000a)((0000)(0017)(ff01)(000a000e000c001d00170018001901000101)(000b0020100)(0023)(0010000e000c02683208687474702f312e31)(00050005010000000)(0033)(002b0009080304030303020301)(000d0018001604030503060308040805080604010501060102030201)(002d00020101)(001c00024001))
```

パディング
あり

```
(0303)(130113031302c02bc02fcca9cca8c02cc030c00ac009c013c01400330039002f0035000a)((0000)(0017)(ff01)(000a000e000c001d00170018001901000101)(000b0020100)(0023)(0010000e000c02683208687474702f312e31)(00050005010000000)(0033)(002b0009080304030303020301)(000d0018001604030503060308040805080604010501060102030201)(002d00020101)(001c00024001)(0015))
```

5.6 結果考察

- 通常状態：どれも変化なし
- BFP対策のみ：ブラウザフィンガープリントのみが変化
- TLS対策のみ：TLSフィンガープリントのみが変化
- 提案手法(BFP対策&TLS対策):

ブラウザ・TLSのフィンガープリント両
方
変化

提案手法の適用により、
フィンガープリンティングを用いたトラッキン
グを

困難にできる可能性がある

6. まとめ

- 現状のフィンガープリンティング対策の多くはブラウザフィンガープリンティングのためのものであり、TLSフィンガープリンティング対策ができていない
 - 両方のフィンガープリントを組み合わせた場合追跡される可能性がある
- 対策として、MITM Proxyを使ったTLSフィンガープリンティング対策を併用することでトラッキングを困難にできる可能性がある

7. 今後の課題

- トラッキング回避のために有用であるフィンガープリントの変更方法を検討
- より簡単に利用するため、ブラウザ自身に TLSフィンガープリンティング対策のための実装を施すべき