Get Rid of Passwords With This One Weird Trick

Understanding the Web Authentication API

DUO LABS

Nick Steele

Senior R&D Engineer

@codekaiju

James Barclay

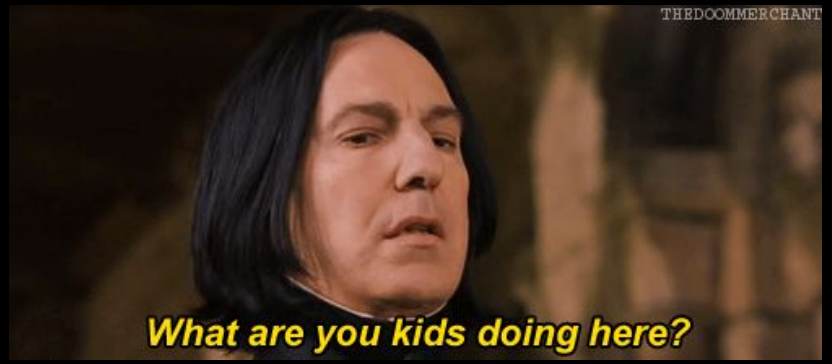
Senior R&D Engineer

@futureimperfect

DUO LABS

Why are we talking about this?

- We're Duo Labs
  - We're the research group at **Duo Security**
  - Strong authentication on the internet is a hard problem
  - We research hard problems!
- We believe that the WebAuthn spec is a good solution to passwordless authentication
- Solving this problem helps pretty much everyone



DUO LABS

Democratization of Security Is Key

- A rising tide lifts all ships
- Solving big security issues together rather than apart
  - Strengthens our community
  - Keeps us honest
- Focus should always be on helping the most users
- Be like Tron

DUO LABS

A Brief History

“In the beginning the password was created. This has made a lot of people very angry and been widely regarded as a bad move.”

- Douglas Adams, sorta

81% of breaches leverage either stolen and/or weak passwords.

Source: 2017 Verizon Data Breach Investigations Report

DUO LABS

Imgur hack: Email addresses, passwords stolen from 1.7M accounts ...
https://www.csoonline.com/.../imgur-email-addresses-and-passwords-stolen-from-17m...
Nov 26, 2017 - Imgur, learning it was hacked in 2014, reacted quickly to notify the public that an **stole** the email addresses and **passwords** for 1.7 million users.

File With 1.4 Billion Hacked And Leaked Passwords Found On The ...
https://www.forbes.com/sites/leemathews/2017/.../billion-hacked-passwords-dark-web/
Dec 11, 2017 - It's also likely that your credentials are listed in a massive file that's floating arour Dark Web. ... Security researchers at 4iQ spend their days monitoring various Dark Web sites, ha forums, and online black markets for leaked and **stolen** data. Their most recent find: a 41 ...

Your passwords are probably a lot worse than you think - CNET
https://www.cnet.com/how-to/find-out-if-your-passwords-been-hacked/
Aug 4, 2017 - Back in May, for example, security research center MacKeeper reported that a mas database of **stolen passwords** had surfaced online. And while it was composed largely of **passw** from a variety of sources, many of them years old, its newfound accessibility -- and conglomera a single ...

Imgur confirms email addresses, passwords stolen in 2014 hack | ZD
www.zdnet.com/article/imgur-reveals-hackers-stole-login-data/
Nov 25, 2017 - (Image: Imgur). Imgur, one of the world's most visited websites, has confirmed a hack dating back to 2014. The company told ZDNe **passwords**, scrambled with the SHA-256 algo of stronger ...

There are 1.9 billion stolen passv
www.businessinsider.com/google-researc
Nov 13, 2017 - Billions of **stolen** user names a internal Google data, researchers found betw a Google search or Gmail account. Gmail, Yah **stolen** ...

DOORDASH
delightful delivery

DoorDash: A $4 billion dollar Food Delivery app has been **hacked**
TechEngage (press release) (blog) - 3 hours ago
4 customers who tweeted their accounts had been **hacked**, told Techcrunch that they used their DoorDash **passwords** for other websites as well ...

ADGUA

Ad Blocker AdGuard Reset All User **Passwords** After Being **Hacked**
Subscription Insider - Sep 25, 2018
AdGuard assured users that the company's servers were not compromised, so the resetting of **passwords** was mostly a preventative measure.

Naked Security

Air Canada **hacked**, user info stolen. If you're a user, change your ...
Boing Boing - Aug 29, 2018
If you're a user, change your **password**. ... did not, however, enjoy the email I received from them this morning warning me they'd been **hacked**.

Vodafone: You used 1234 as your password and were hacked? You cover the cost

Updated: Hackers are behind bars for stealing $30,000 from accounts, but Vodafone wants their victims to pay the tab.

By Charlie Osborne for Zero Day | September 6, 2018 -- 08:14 GMT (01:14 PDT) | Topic: Security

DUO LABS


ALL YOUR FAULT!

“Kind of a nightmare...”

- The guy who invented it

DUO LABS

“Passwords Suck”

- Most People

“Pas ck”

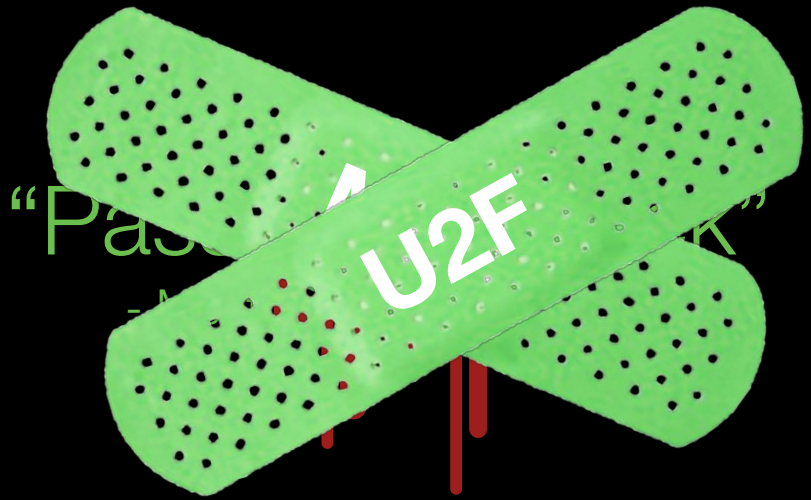
- Most Peop

DUO LABS

Multi Factor Authentication

- Better than only 1st factor, but…
  - SecureID tokens can be stolen
  - HOTP and TOTP passwords can still be intercepted
  - Mobile device could be stolen, compromised
  - SIM could be cloned, emails could be compromised
  - SMS/email could be delayed/dropped, preventing login

DUO LABS



DUO LABS

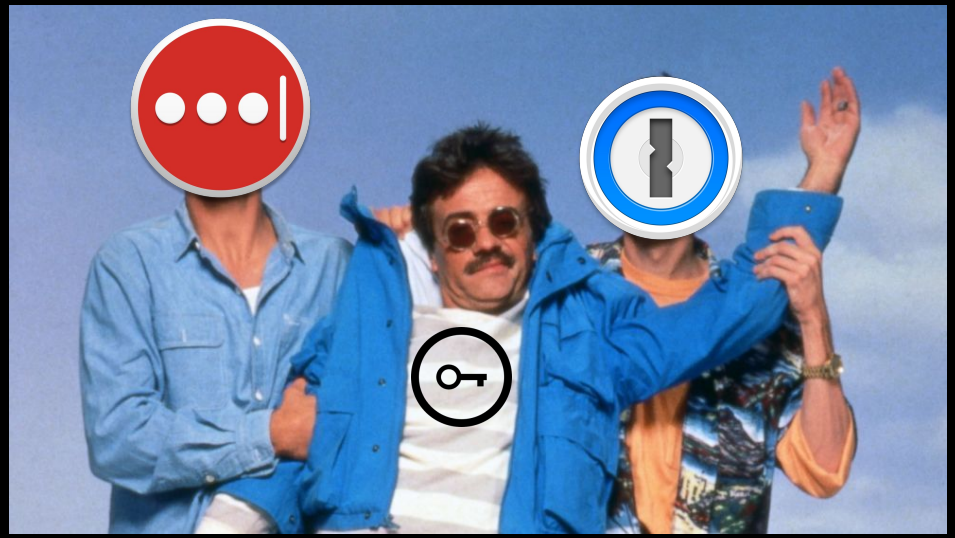
Universal Second Factor

- Better than only 1st factor and traditional MFA but…
  - Requires physical token (usually)
  - Tokens can be expensive ($19 and up)
  - Hard to use (if even possible) on mobile devices
  - Isn't supported natively in most browsers (not really universal…)
  - Hard to convince people to use it casually

DUO LABS

“The average… user has over 107 accounts registered to one email address… In 2020, the average number of accounts per internet user will be 207”

- Dashlane, 2015



DUO LABS

Meanwhile… in the year 2015

- Phones are becoming smarter
  - Most have a security module, like a TEE or SEP
    - Capable of handling complex cryptographic operations
  - Biometric authentication is common on these devices
  - 77% of Americans own a smartphone in 2017 (68% in 2015)
- FIDO drafts Universal Authentication Factor Spec
  - Spec describes a method for authenticating users via mobile device to web applications using credential keys created by the phone, (and authenticated with a biometric)
  - Not a lot of traction, but paved the way for…

Web Authentication

DUO LABS

Web Authentication

DUO LABS

WebAuthn

DUO LABS

WebAuthn is…

"...an API enabling the creation and use of strong, attested, scoped, public key-based **credentials** by web applications, for the purpose of strongly authenticating users."

DUO LABS

WebAuthn

"...[with WebAuthn] one or more public key credentials, each scoped to a given **Relying Party**, are created and stored on an **authenticator** by the user agent in conjunction with the web application."

DUO LABS

Credentials

Strong

Attested

Scoped

DUO LABS

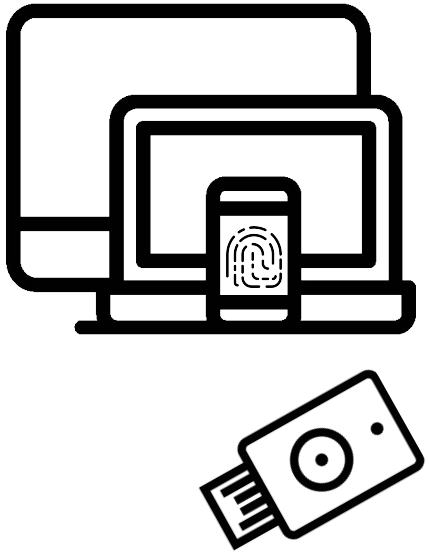
Relying Party

- AKA The website that is requesting credentials
- Credentials for the Relying Party are bound by origin (scoped!)
  - Possible to use for subdomains, like `sub.example.com`, but not vice versa
- Cannot talk directly to the authenticator or (by default) identify the authenticator
- A breach of the Relying Party's credential database would be moot

DUO LABS

Authenticators

- Capable of creating strong and secure key pairs
- In the case of most mobile devices, includes a biometric
-

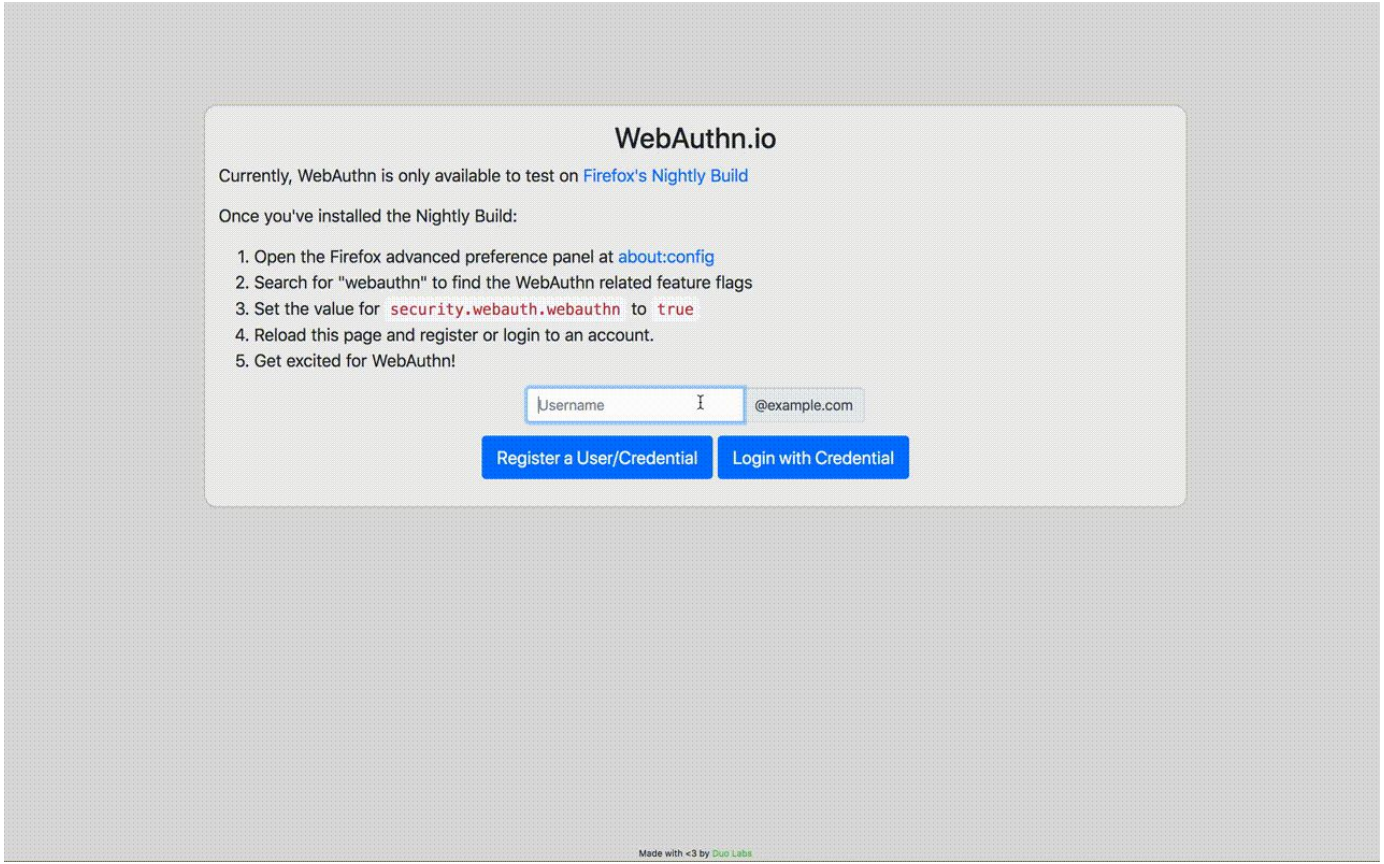


DUO LABS

Why is WebAuthn important?

- Raises the bar for

DUO LABS


WebAuthn.io
Currently, WebAuthn is only available to test on Firefox's Nightly Build
Once you've installed the Nightly Build:
1. Open the Firefox advanced preference panel at about:config
2. Search for "webauthn" to find the WebAuthn related feature flags
3. Set the value for security.webauth.webauthn to true
4. Reload this page and register or login to an account.
5. Get excited for WebAuthn!
Username
@example.com
Register a User/Credential
Login with Credential
Made with <3 by Duo Labs

DUO LABS

What we did

DUO LABS

FIDO2

- In March the FIDO Alliance introduced The FIDO2 Framework
  - WebAuthn + CTAP2

What Does That Even It Mean? Cont.

- **Unique credentials per relying party (website)**
- **Can’t (easily) use WebAuthn to track users**
- **The browser/platform/user agent is responsible for communicating with the authenticator**
- **We place some trust in authenticators**
  - This trust is bootstrapped via attestation
- **User communicates with the authenticator to authorize a signing operation**
  - No secrets are ever sent to the server

How WebAuthn Handles It

- **User talks to device instead of directly to server**
- **Biometric, PIN, gesture, etc. are used to authorize the signing operation**
- **Challenge is sent to the user by the relying party**
  - Browser/platform asks the authenticator to perform the signing operation
  - Trust is placed in the authenticator to protect keys, (bootstrapped by attestation)

Show different authenticator possibilities, (token, phone, laptop with TPM, etc.). maybe show the login flow for UAF and talk about how much better that is than typing passwords. Platform (SGX, TEE, TPM, SEP) vs. Cross-Platform authenticators, (U2F, Biometrics)

Some Quick Background on Webauthn

- Spec draft started in May 2016 (Picked up Jan 2017)
- Strongly resembles the Universal Authentication Factor Spec
  - Some of the same authors as well
  - UAF is like WebAuthn but for native applications running on mobile devices.
  - Knowledge of this spec isn't necessary for WebAuthn
- Includes contributors from Google, Microsoft, Nok Nok, and Mozilla
- FIDO Alliance have already given a workshop on implementing WebAuthn
- Yubico and Feitian (U2F and HSM manufacturers) are leading the pack in providing support for the spec at the hardware.
  - WebAuthn will allow for native U2F support, currently only available in Chrome.

Implementation Roadmap

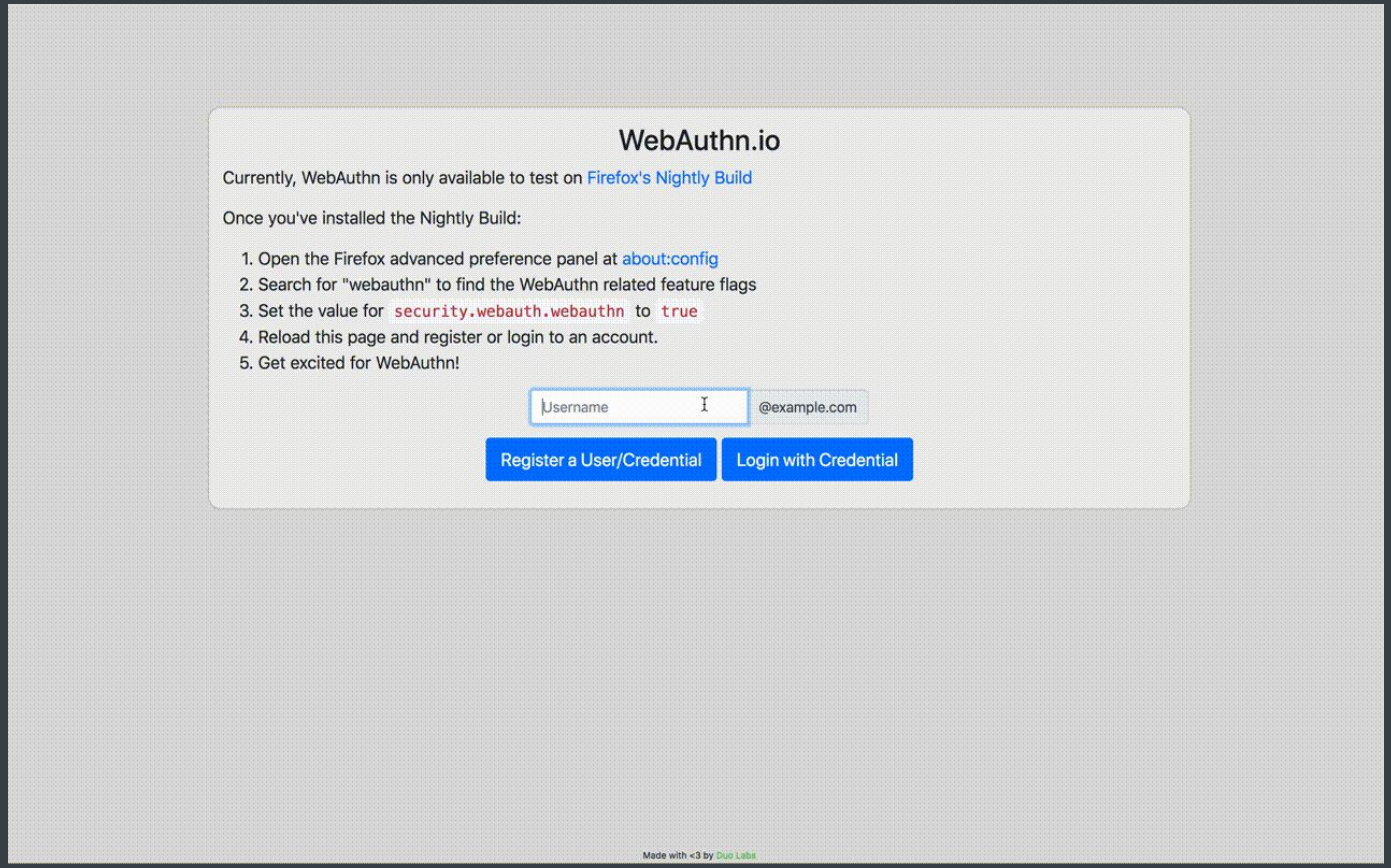
- W3C Candidate Recommendation coming soon
- In active development for Gecko, Blink, and Chakra
- Working behind a flag in
  - Chrome Canary
  - Firefox Nightly
- Slated for stable build release in
  - Chrome version 65 (now at 64)
  - Firefox version 60 (now at 58)
- If you work at Microsoft please give me access to the working Edge build
  - I know you have it and I will buy you a beer
- Apple is Apple
  - Probably making their own spec lol
  - Nah jk they’re part of the W3C

What We Did (Last Summer)

- Started researching the spec in September
- Wrote the first public working PoC of WebAuthn
  - webauthn.io
- Wrote another PoC of WebAuthn in Python
- Met with the authors and contributed to the spec
- Spread the WebAuthn gospel

DUO

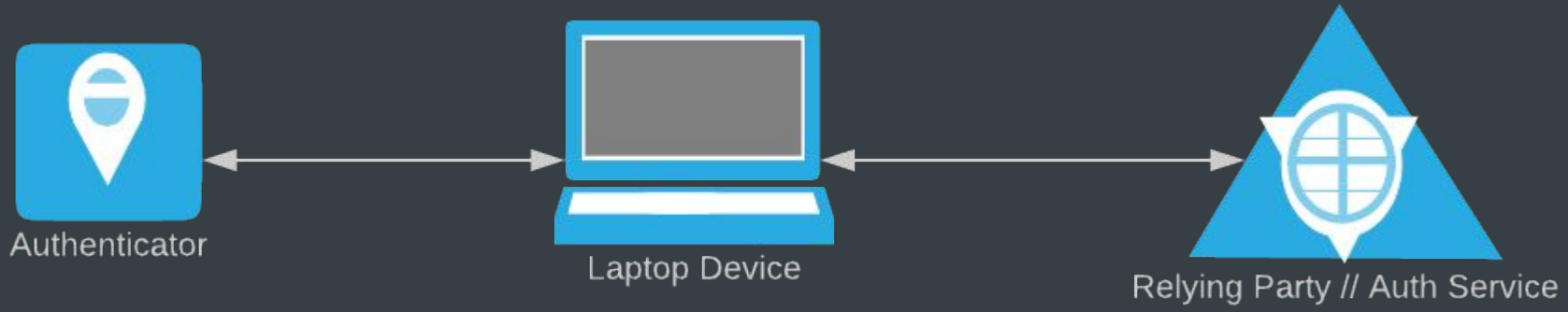
Show and Tell


WebAuthn.io
Currently, WebAuthn is only available to test on Firefox's Nightly Build
Once you've installed the Nightly Build:
1. Open the Firefox advanced preference panel at about:config
2. Search for "webauthn" to find the WebAuthn related feature flags
3. Set the value for security.webauth.webauthn to true
4. Reload this page and register or login to an account.
5. Get excited for WebAuthn!
Username
@example.com
Register a User/Credential
Login with Credential
Made with <3 by Duo Labs

Now, Before We Get Technical...

- There are only two main WebAuthn operations, referred to as ceremonies.
  - Ceremonies are an attempt to extend the concept of a network protocol to include human interaction. What is out-of-band to a protocol is in-band to a ceremony.
- These ceremonies involve 4 components:
  - A User
  - An Authenticator (Yubikey, Laptop TPM/TEE, Android TPM, etc)
  - The Client (Browser)
  - A Relying Party (The website or application the user is accessing)
    - The Relying Party can act through an Authentication Server
    - To keep it simple, we'll concat the two.

Registration Ceremony



The user creates a strong credential key pair verified and stored in part by the Relying Party

WebAuthn Authenticators

If Not Passwords, Then What?

- User Verification
  - Phone
    - iOS (Secure Enclave Processor)
    - Android (Trusted Execution Environment)
  - Laptop or Desktop
    - Windows Hello (Face or PIN to unlock keys stored in TPM)
    - macOS (Touch ID to unlock keys stored in SEP)
  - External Authenticator, (e.g., FEITIAN BioPass)
- User Presence
  - U2F (only user presence is verified, UV bit set to 0)

The prevalence of consumer security hardware in recent years makes these authenticators possible. Phones, laptops, and various other consumer electronic devices are shipping with TEEs/SEPs/TMPs/etc. these days.

WebAuthn Flow

tbh this is UAF, but WebAuthn is very (very) similar.

How Does WebAuthn Work?

- Credential management API stuff
  - WebAuthn extends this

- Authentication Server
- Relying Party
- Authentication Server and RP can be the same server or separate
- CTAP (Client to Authenticator Protocol)
- Attestation
  - Attestation types, (fido-u2f, none, tpm, SafetyNet, etc.).

WebAuthn Relying Party Operations

- Stuff stuff stuff

How does WebAuthn work for Non U2F?

- A User goes to a URL owned by a Relying Party to and is prompted to create a new account. They hit “Register with your Phone”
- Their browser or an app opens on their phone with a prompt to Register an account with the site that they are on.
- An account is created and they are let into the site.
- That’s it!
- Subsequent login requests (called Auths or Assertions) follow a similar pattern.

But how does WebAuthn work?

- User Client, Relying Party, and Auth Server
  - The RP can also be the Auth Server
- There are two main Auth Ceremonies: Registration and Assertion
  - `navigator.credentials.create({credentialOptions})`
  - `navigator.credentials.get({credentialID&Data})`
- The information we receive back from these ceremonies is sent to the Auth Server for validation

CTAP

- Specification for the application layer protocol for communication between a client and external authenticators
  - Handles the requirements for transport over USB, NFC, and BLE
- Non-external authenticators don’t have to conform to CTAP.
  - Aims at standardizing the communication between a laptop and a smartphone.
- Not necessary to know CTAP to understand Webauthn

WebAuthn Registration

Registration

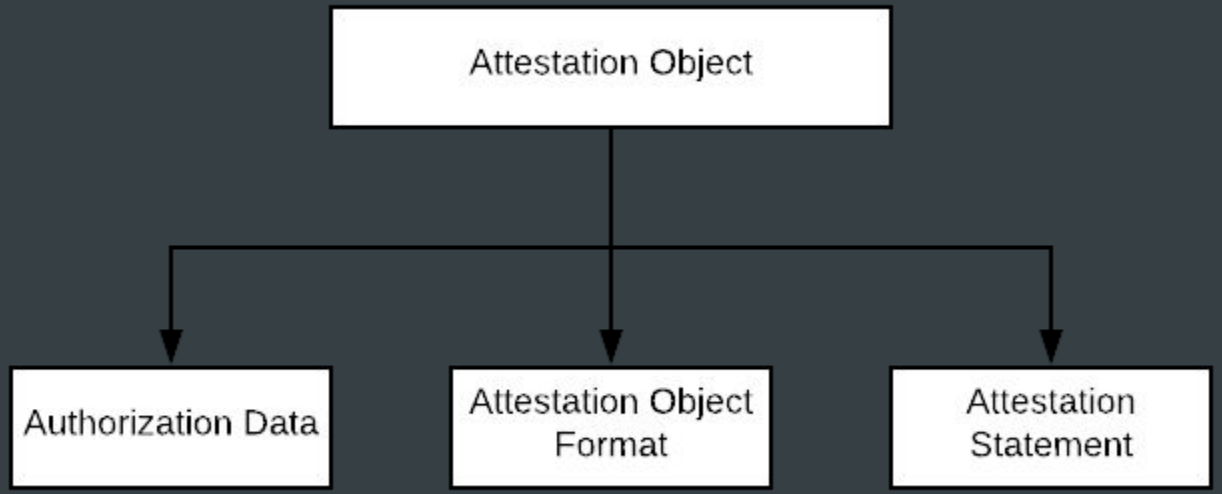
- The RP provides the User Client with Credential Options for accessing an authenticator via WebAuthn. The RP can request specific authenticator traits (must support public key creation, must make RSA/EC key type).
- We get a signed response from the authenticator that proves what type of authenticator it is (in an Attestation Object) and we get a Public Key (EC) specific to the Relying Party.
- We can verify facets of the authentication ceremony, like if the authenticator proves User Presence AND User Verification.
- We store a Credential ID and The Public Key (and other relevant info).

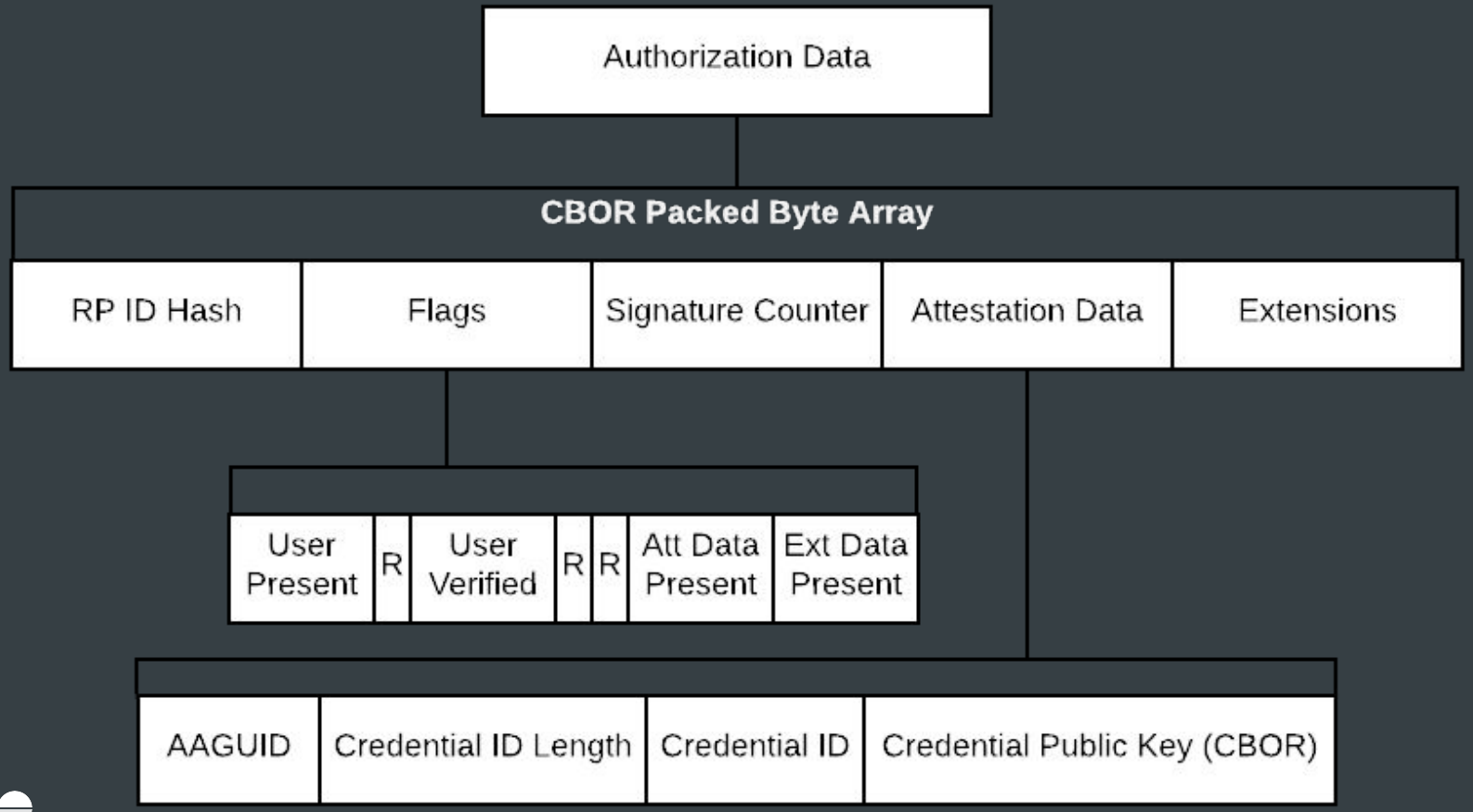
Verifying the User

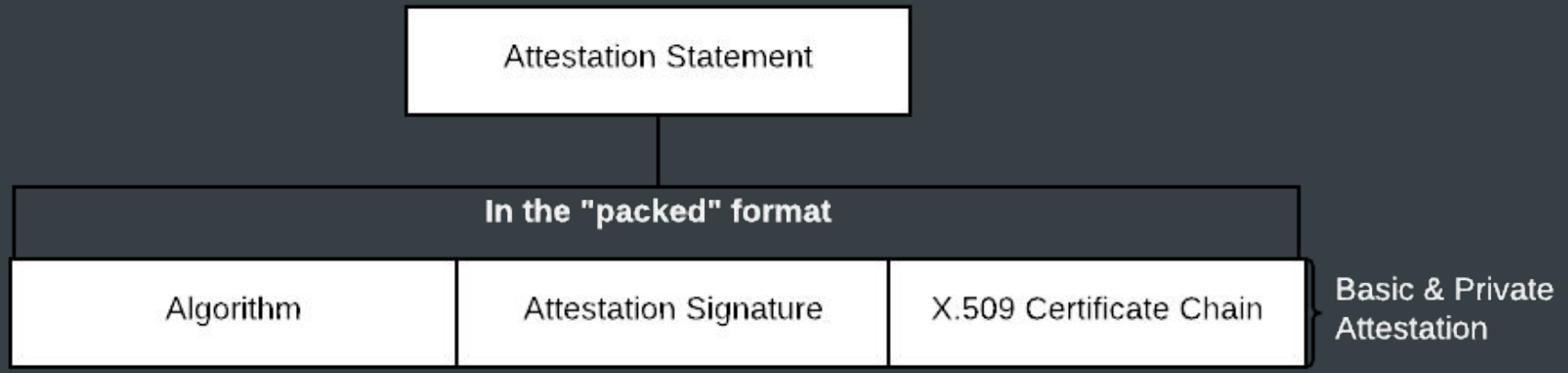
- The RP provides the User Client with Credential Options for accessing an authenticator via WebAuthn. The RP can request specific authenticator traits (must support public key creation, must make RSA/EC key type).
- We get a signed response from the authenticator that proves what type of authenticator it is (in an Attestation Object) and we get a Public Key (EC) specific to the Relying Party.
- We can verify facets of the authentication ceremony, like if the authenticator proves User Presence AND User Verification.
- We store a Credential ID and The Public Key (and other relevant info).

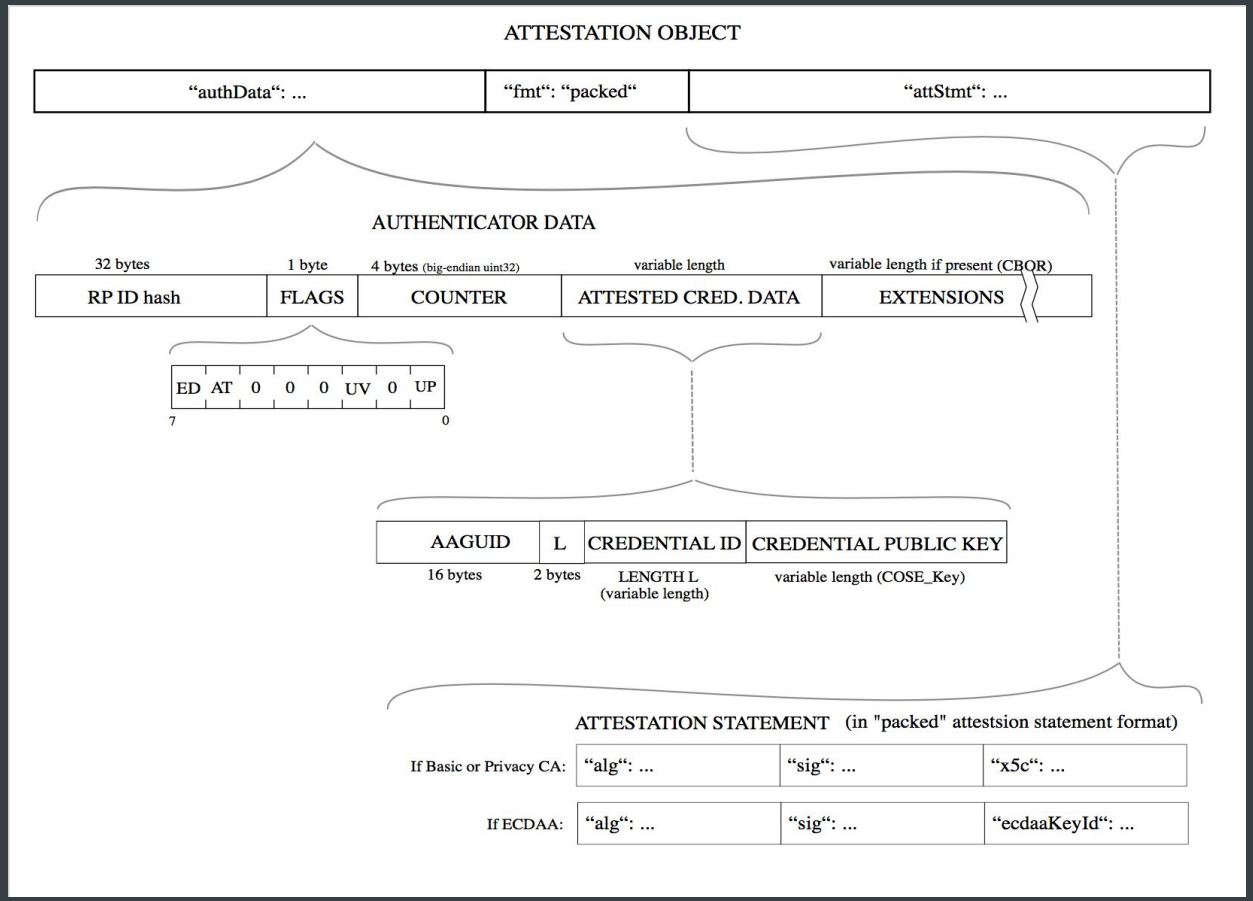
Verifying the Authenticator

- The RP provides the User Client with Credential Options for accessing an authenticator via WebAuthn. The RP can request specific authenticator traits (must support public key creation, must make RSA/EC key type).
- We get a signed response from the authenticator that proves what type of authenticator it is (in an Attestation Object) and we get a Public Key (EC) specific to the Relying Party.
- We can verify facets of the authentication ceremony, like if the authenticator proves User Presence AND User Verification.
- We store a Credential ID and The Public Key (and other relevant info).


Attestation Object
Authorization Data
Attestation Object Format
Attestation Statement


Authorization Data
CBOR Packed Byte Array
RP ID Hash
Flags
Signature Counter
Attestation Data
Extensions
User Present
R
User Verified
R
R
Att Data Present
Ext Data Present
AAGUID
Credential ID Length
Credential ID
Credential Public Key (CBOR)


Attestation Statement
In the "packed" format
Algorithm
Attestation Signature
X.509 Certificate Chain
Basic & Private Attestation


ATTESTATION OBJECT
“authData“: ...
“fmt“: “packed“
“attStmt“: ...
AUTHENTICATOR DATA
32 bytes
1 byte
4 bytes (big-endian uint32)
variable length
variable length if present (CBOR)
RP ID hash
FLAGS
COUNTER
ATTESTED CRED. DATA
EXTENSIONS
ED AT 0 0 0 UV 0 UP
7
0
AAGUID
L
CREDENTIAL ID
CREDENTIAL PUBLIC KEY
16 bytes
2 bytes
LENGTH L
(variable length)
variable length (COSE_Key)
ATTESTATION STATEMENT (in "packed" attestsion statement format)
If Basic or Privacy CA:
“alg“: ...
“sig“: ...
“x5c“: ...
If ECDAA:
“alg“: ...
“sig“: ...
“ecdaaKeyId“: ...

WebAuthn Assertion (Login)

Assertion

- Pass back the Credential ID we got during registration so the Authenticator knows what credential we want.
    - We can look for specific authenticators and traits here as well (i.e. we want credentials with only these IDs, and only receive them over bte, nfc, or usb, etc)
- We get a response back from the authenticator with some data and a signature generated from that data and signed by the Credential's Private Key.
- We use the Public Key stored on Registration to verify that the signature is valid and check that the data conforms to our Relying Party's policies.

Asserting ownership of a Credential

- The RP provides the User Client with Credential Options for accessing an authenticator via WebAuthn. The RP can request specific authenticator traits (must support public key creation, must make RSA/EC key type).
- We get a signed response from the authenticator that proves what type of authenticator it is (in an Attestation Object) and we get a Public Key (EC) specific to the Relying Party.
- We can verify facets of the authentication ceremony, like if the authenticator proves User Presence AND User Verification.
- We store a Credential ID and The Public Key (and other relevant info).

WebAuthn Extensions

- Stuff stuff stuff

WebAuthn Shortcomings

- WebAuthn is only as strong as the weakest link
  - If email is your recovery mechanism when a user loses their authenticator, that's your bottleneck
- Multiple authenticators
  - For privacy reasons, an RP can't know whether an authenticator is present on a given device until asking
  - Do you prompt for a password, or for the user to log in with their phone/laptop?

What's next?

- Spec is in development for Chrome, Firefox, and Edge
  - Currently behind flags in
- Why is this more likely to catch on than FIDO's UAF Standard?
  - Slightly lower barrier to entry - WebAuthn built in to browsers, but still requires implementers to handle parsing/verifying the Client's responses (Duo could help!)
  - More Vendors involved - Webauthn Spec is being authored by broader group. Android SafetyNet planned support is a good signal. Google already uses the Credentials API to create private and federated passwords
  - FIDO UAF Working group now references WebAuthn Work

Authn

github.com/duo-labs/py_webauthn

WebAuthn Implementations

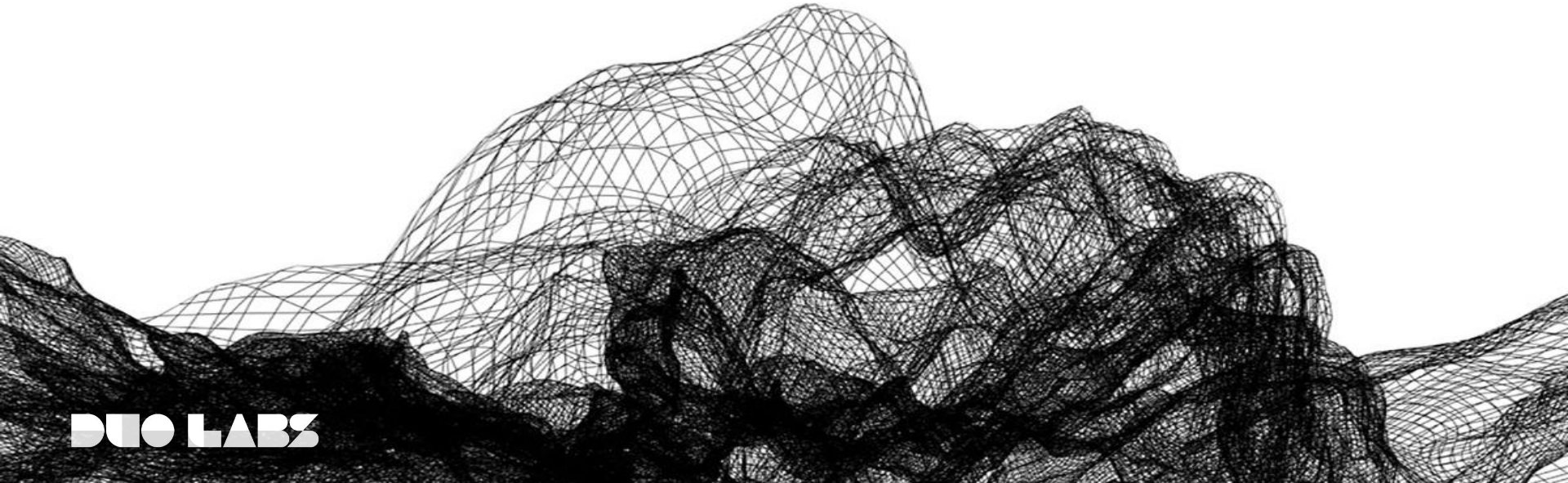
Author	Repository	Works?
Google	webauthndemo	Yes
FIDO	webauthn-demo	Yes
Duo Labs	webauthn	Yes
Duo Labs	PyWebAuthn	Yes

Key Take-Aways

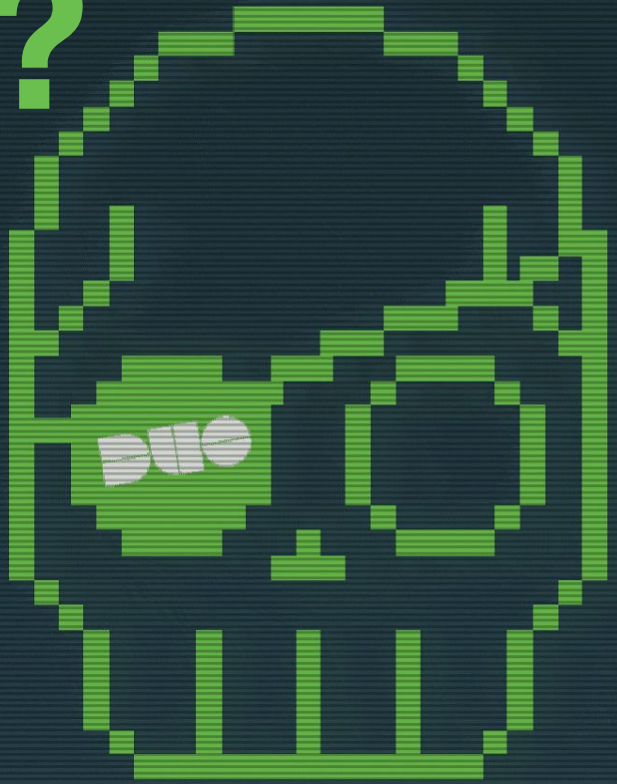
- Passwords aren’t enough
- WebAuthn is a new standard for using public-key credentials on the web, for the purpose of authenticating users
- WebAuthn isn't perfect, but it's our current best bet for replacing passwords on the web
- Major browser vendors and platform owners are investing time/money/resources into WebAuthn, so expect to hear more in the coming months/years

Questions?

@codekaiju & @futureimperfect



Questions?



@codekaiju && @futureimperfect
nsteele@duo.com && jbarclay@duo.com
github.com/duo-labs/webauthn