# Divergences of Ed25519 in Web Crypto and beyond

## Inconsistencies in implementations and specifications

CFRG - IETF 121

# CFRG Curves in Web Crypto

- Secure Curves in the Web Cryptography API draft specifies {Ed,X}{25519,448}

| Implementor | Ed25519 | X25519 | Ed448 | X448 | in a stable release |
|---|---|---|---|---|---|
| Chromium (Chrome, Edge)[1] | ✓ | ✓ | | | |
| WebKit (Safari)[2] | ✓ | | | | ✓ |
| Gecko (Firefox)[3] | ✓ | ✓ | | | ✓ |
| Node.js[4] | ✓ | ✓ | ✓ | ✓ | ✓ |
| Deno[5] | ✓ | | | | ✓ |
| Netlify Edge Functions[5] | ✓ | | | | ✓ |
| Bun[6] | ✓ | | | | ✓ |
| Cloudflare Workers[7] | ✓ | ✓ | | | ✓ |
| workerd[7] | ✓ | ✓ | | | ✓ |
| Vercel's Edge Runtime[8] | ✓ | ✓ | | | ✓ |
| Vercel's Edge Functions[8] | ✓ | ✓ | | | ✓ |
| Flow[9] | ✓ | ✓ | ✓ | ✓ | ✓ |

# Inconsistencies in Ed25519 implementations

- Unfortunately each browser's implementation is different
- Chrome wants to do what existing implementations do
- Chrome also wants a fully "locked down" specification
- Firefox and Safari want to improve upon RFC 8032 (in different directions)
- Would be nice to be consistent with CFRG

# Randomized signatures

- Apple generates randomized signatures, presumably as per draft-irtf-cfrg-det-sigs-with-noise
  - (Not only in Safari; all applications using CryptoKit)
- Everyone else generates deterministic signatures as per RFC 8032

> ISSUE 28: Allow creating randomized EdDSA signatures?
>
> Some implementations may (wish to) generate randomized signatures as per draft-irtf-cfrg-det-sigs-with-noise instead of deterministic signatures as per [RFC8032].

# Small-order checks

- Firefox wants to reject small-order points
  - Seems Legit: Automated Analysis of Subtle Attacks on Protocols that Use Signatures
  - The Provable Security of Ed25519: Theory and Practice
- Chrome wants to do whatever BoringSSL does (currently no small-order checks)

2. If the key data of *key* represents an invalid point or a small-order element on the Elliptic Curve of Ed25519, return `false`.

> ISSUE 27: Make small-order checks in EdDSA optional?
>
> Not all implementations perform this check.

3. If the point R, encoded in the first half of *signature*, represents an invalid point or a small-order element on the Elliptic Curve of Ed25519, return `false`.

> ISSUE 27: Make small-order checks in EdDSA optional?
>
> Not all implementations perform this check.

# Verification equation

- Everyone seems to do unbatched verification

4. Perform the Ed25519 verification steps, as specified in [RFC8032], Section 5.1.7, using the cofactorless (unbatched) equation, `[S]B = R + [k]A'`, on the *signature*, with *message* as *M*, using the Ed25519 public key associated with *key*.

- Thus: RFC 8032 is overly flexible?

```
3.  Check the group equation [8][S]B = [8]R + [8][k]A'.  It's
    sufficient, but not required, to instead check [S]B = R + [k]A'.
```

# Questions

- Move draft-irtf-cfrg-det-sigs-with-noise towards an RFC?
- Document to allow or recommend small-order checks?
- Or: RFC 8032bis?
  - Allow randomized signatures
  - Allow or recommend small-order checks
  - Mandate or recommend unbatched verification?

# Thoughts? Questions?