Credential Manager Trust Groups

Explainer explainer

github.com/w3c/webauthn/blob/main/explainers/credential-manager-trust-group-key.md

1. What problem are we solving?
2. RPK extension structure & usage
3. Implementation ideas for providers

Passkey Trust Signals

- What if the provider backend was hacked?
- What if the provider is lying about their security?
- Is the key device-bound? (I'm regulated)
- What if the provider account itself was phished?

DC API / DBSC help solve

Bigger problem at scale

Opportunities

Providers generally have robust sign in policies.

Providers can afford to ask more of the user than general RPs.

→ Let's carry this forward to relying parties.

CMTG at 10k feet

CMTG allows RPs to know
when credential managers protect
themselves against phishing

Constraints

Managers cannot always do unphishable sign-in.

Proved absence of a phisher is *not a property of one sign-in*.

Constraints

Managers cannot always do unphishable sign-in.

Proved absence of a phisher is *not a property of one sign-in*.

Constraints

Managers cannot always do unphishable sign-in.

Proved absence of a phisher is *not a property of one sign-in*.

Device & session trust

Idea: Mangers should convey the goodness of a device.

Nearly impossible.

- "Goodness" of a device is very RP specific
- Managers would have to attest "this is a legit device", but no good answer on multiple large platforms.

CMTG at 1000 feet

1. Let RPs assess session trust when needed.
2. Tell RPs when one session is *"unphishably" related* to another.
3. . . .
4. Profit

```
const cred = await navigator.credentials.get({
  publicKey: {
    challenge: ...,
    extensions: {
      cmtgKey: true  /* also works for .create */
    }
  }
});
```

Result will always have:

1. An extra public key *CMTGK*
2. Proof of possession of the corresponding secret key

Device A

Device A

Device B

Device A

Device B

Device C

Device A

Device B

Device C

Details

- Each CMTG keypair belongs to exactly one passkey (and thus RP)
- Each passkey may have multiple CMTGKs
- Each device may have relationships to multiple other devices, only one CMTGK is presented on each operation

Important: Devices never have CMTG private keys they aren't supposed to.

Admissible signals

Deliberately not defined by W3C proposal.
Idea:

- FIDO (CPSIG?) sets *guidelines* for signals that allow syncing RPKs.
- Describe intent, not a definitive list. (Examples and counterexamples ok.)

Strawman starting point:

- Provider verified proximity of devices. E.g. with cross-device passkey
- Provider used same strong physical factor on two devices, e.g. security key
- Provider verified strong phishing resistant external factor, e.g. phone number

Relying party use

1. Ask for CMTG.
2. If the CMTG key is unknown, consider other trust signals
3. Record trust signals, keyed on the device + the CMTG key seen

Later:

1. Ask for CMTG
2. Look up CMTG key in "trust database".
3. Consider trust signals from *other devices* that presented the same CMTG key.

Some notes on possible implementation

Manager maintains internal keys, one per device.
Shared with sync server, distributed to other devices **if** a relationship exists.

First request for CMTG from RP:

1. Select passkey.
2. Since no associated CMTG exists, create one.
   Store in sync but encrypted with this device's internal key.

Later

1. Select passkey.
2. Load encrypted CMTG keys from sync.
   If any can be decrypted with any internal key, use it.
   If not, create a new one, encrypt with this device's internal key.