VICEROY: GDPR-/CCPA-compliant Verifiable Accountless Consumer Requests

Scott Jordan[1], Yoshimichi Nakatsuka[1], Ercan Ozturk[1], Andrew Paverd[2], Gene Tsudik[1]

[1] University of California, Irvine
[2] Microsoft

Viceroy butterfly
https://unsplash.com/@jcotten

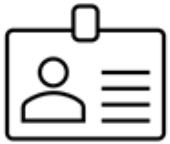
Data Protection Regulations

- GDPR (General Data Protection Regulation)
  - *data subjects* in the EU/EEA
- CCPA (California Consumer Privacy Act)
  - *consumers* who are California residents
- …
- Grant consumers legal rights over their data:
  - Access
  - Correct
  - Delete

Verifiable Consumer Request (VCR)

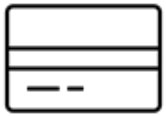
- Request from a **consumer** to a **service provider** (e.g., website) to access/modify/delete personal data
- Website must **verify** authenticity of request
  - Otherwise, there are privacy consequences
- Verification is straightforward when consumer has an account
  - Ask the consumer to log in etc.
- But what about consumers without accounts?
  - Data protection regulations still apply

How are “Accountless” consumers currently verified?



Government-issued ID

Signed statement



Credit card number

Phone interview

Ad-hoc, Insecure, Privacy-invasive

Why not use Cookies  ?

When submitting a VCR, accountless consumers could authenticate themselves by providing the unique cookies they received when accessing the service.

Pros:

- Privacy-preserving

} *Server learns nothing new*

Cons:

- Cookies might not be unguessable
- Must securely sent over the network
- Require secure storage

} *Cookies are **symmetric** authentication tokens*

Introducing VICEROY

A framework enabling **accountless** consumers to request their data in a **secure** and **privacy preserving** manner.

Specifically, VICEROY…

- allows consumers to generate VCRs without relying on symmetric tokens,
- allows website operators to efficiently and securely verify VCRs,
- can be integrated into existing websites with minimal changes.

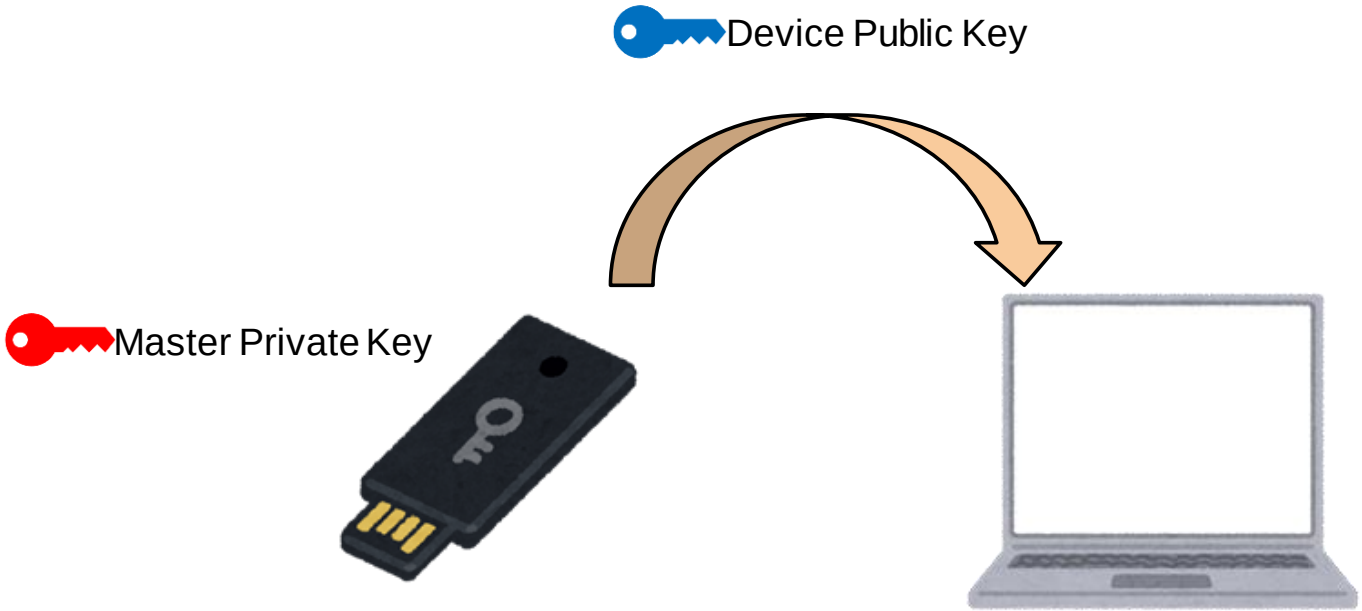
Overview of VICEROY

Trusted Client Device

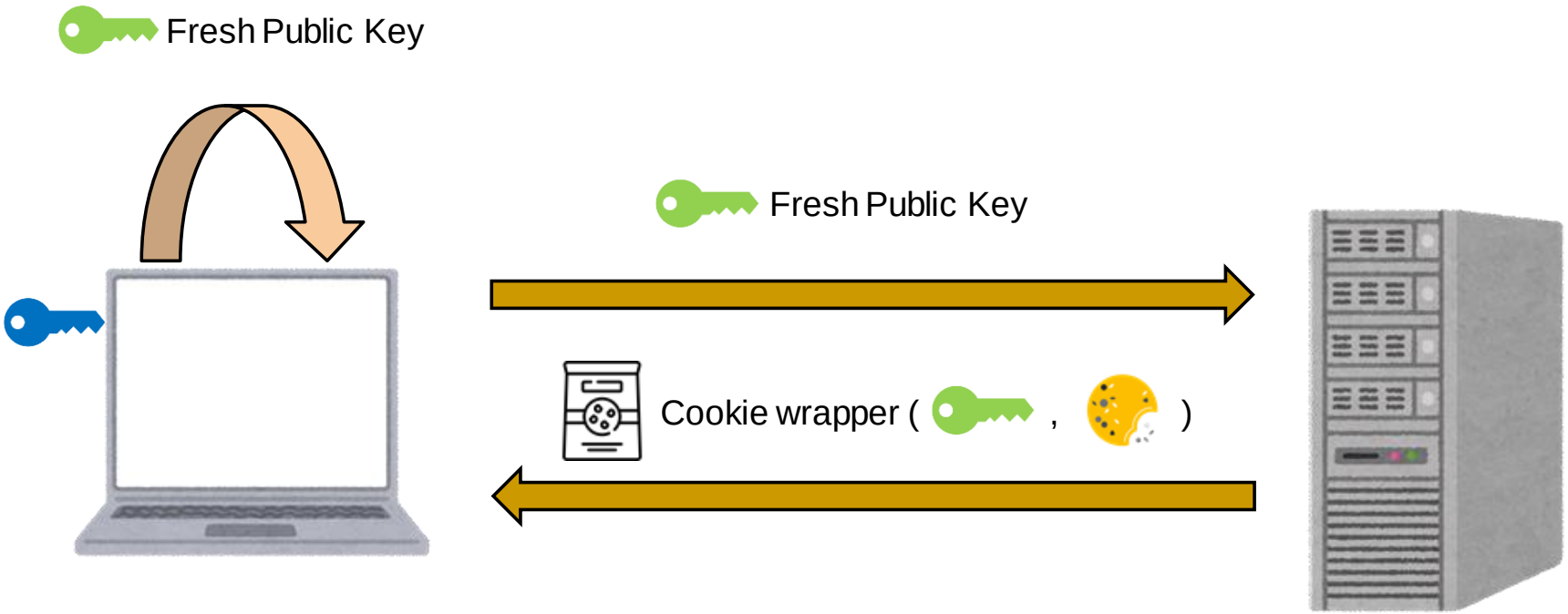
Client Device

Server

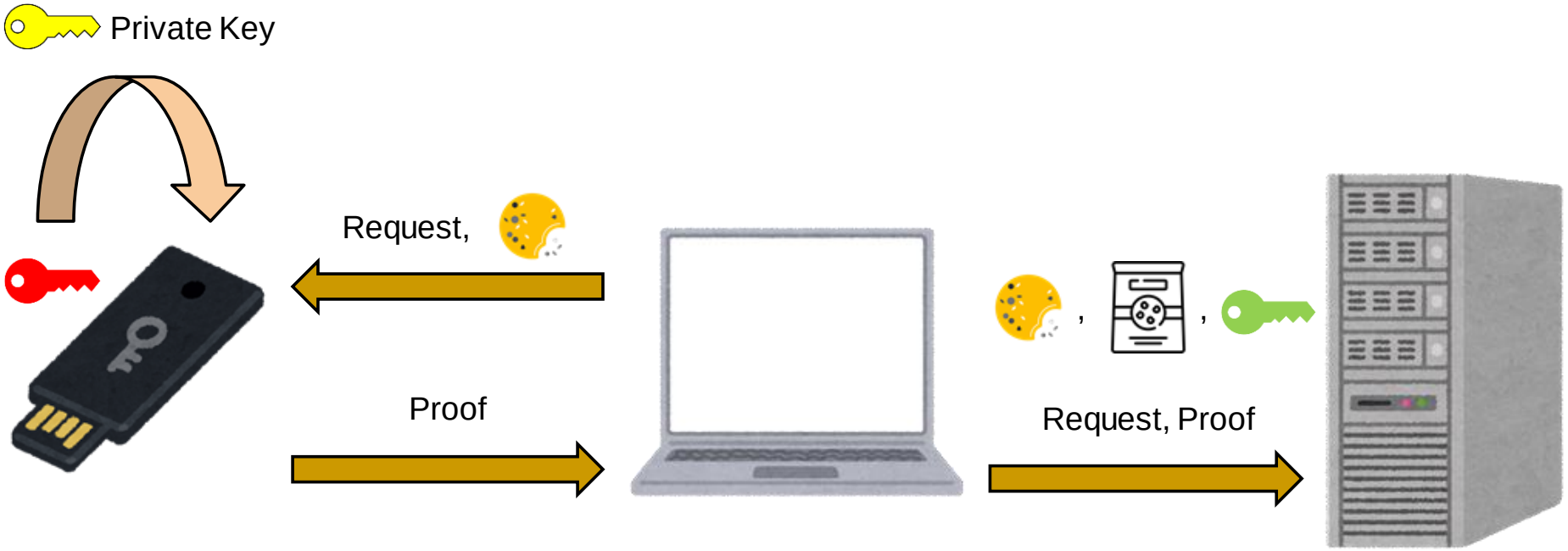
1. Setup phase



2. Visiting a website



3. Proving data ownership

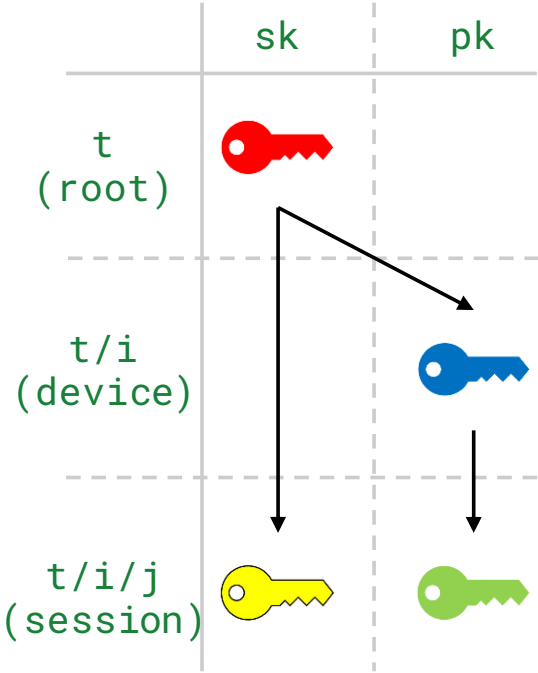


Key management

Challenge: Will the user have to store a large number of cryptographic keys (e.g., one per session)?

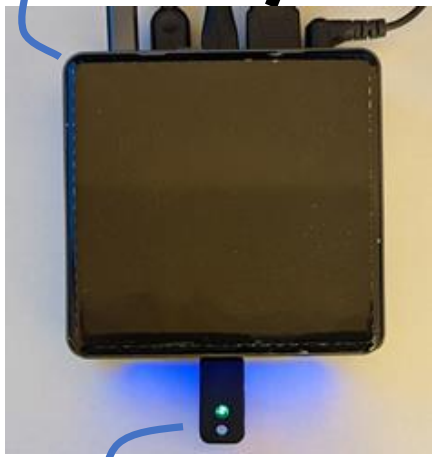
Solution: Use Derivable Asymmetric Keys (e.g., BIP32)

- User only has to store one master private key
- Master private key can be stored offline on trusted client device
- Client can use multiple devices rooted in a single trusted device

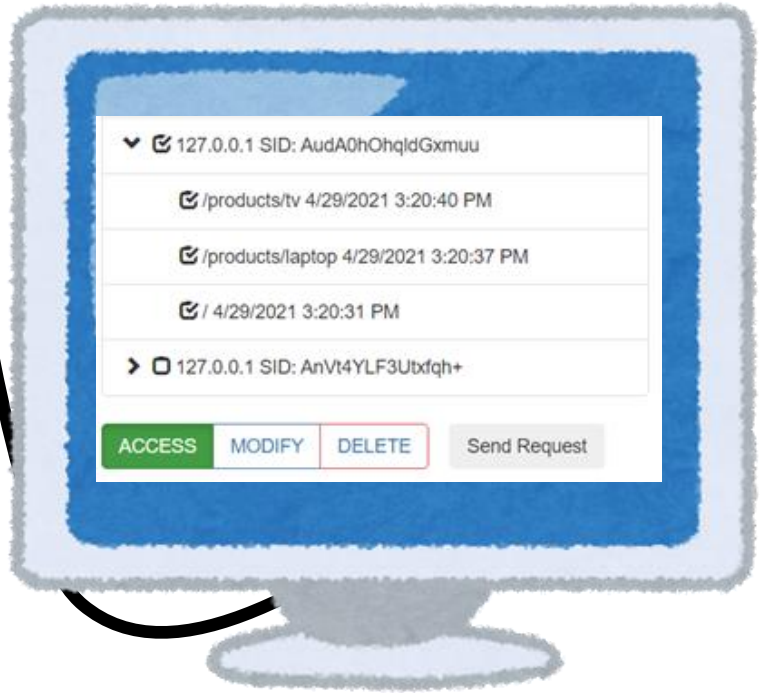


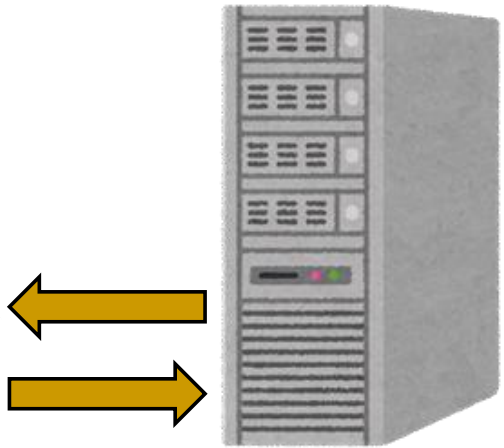
Implementation

Consumer Device
(Native Application)



Trusted Consumer Device
(Solokey)





Security Analysis

- Unforgeability
  - Only the client who originally interacted with server can create a valid VCR.
- Replay resistance
  - A server will accept a valid VCR at most once.

Trace properties verified using the Tamarin Prover

- Unlinkability
  - Honest-but-curious server should be unable to link a VCR to a specific client, or to link multiple VCRs to the same client.

Non-trace property verified through manual inspection

Evaluation (Latency)

Browsing a new web page:

Key Derivation	Wrapper Generation	Wrapper Verification	Wrapper Storage
24.6 ms	0.4 ms	18.8 ms	6.5 ms

Generating and verifying a VCR:

	VCR Generation	VCR Verification
VCR Flow	1357.4 ms	1.5 ms

Evaluation (Data transfer & Storage)

Data transfer:

	Request	Response	Total
Obtain Wrapper	0.72 kB	0.38 kB	1.10 kB
Issue VCR	0.99 kB	0.28 kB	1.27 kB

Client-side storage requirement:

- One year of VICEROY usage: **23 MB**

Open research questions

- Linking through metadata
  - Hide IP address, use TEE/PIR when retrieving data
- Shared devices (a.k.a., roommate problem)
  - For example, two people watching smart TV
  - Who “owns” the collected data?
- 3rd party cookie support
  - How to extend VICEROY to allow clients exercise their rights over data collected by 3rd parties (e.g., advertising networks)?
- New business opportunity: Cookie wrapper storage
  - Consumers may need to store their cookies and wrappers indefinitely
  - Cookies and wrappers can be stored by external service
  - “VCR as a service”?

Conclusion

VICEROY -
A privacy-preserving and scalable framework for producing *proofs of data ownership*

Designed to support multiple devices, Web environment, and long-term use

Proof of concept implementation that is easy to integrate into browsers

Latency, bandwidth, and storage evaluation show VICEROY is efficient

Implementation & Tamarin proof available online: https://github.com/sprout-uci/VICEROY →

Questions? Email: nakatsuy@uci.edu

Appendix

Verifiable Consumer Request (VCR)

With account

- Standard
- Secure

Verifiable Consumer Request (VCR)

With account

- Standard
- Secure

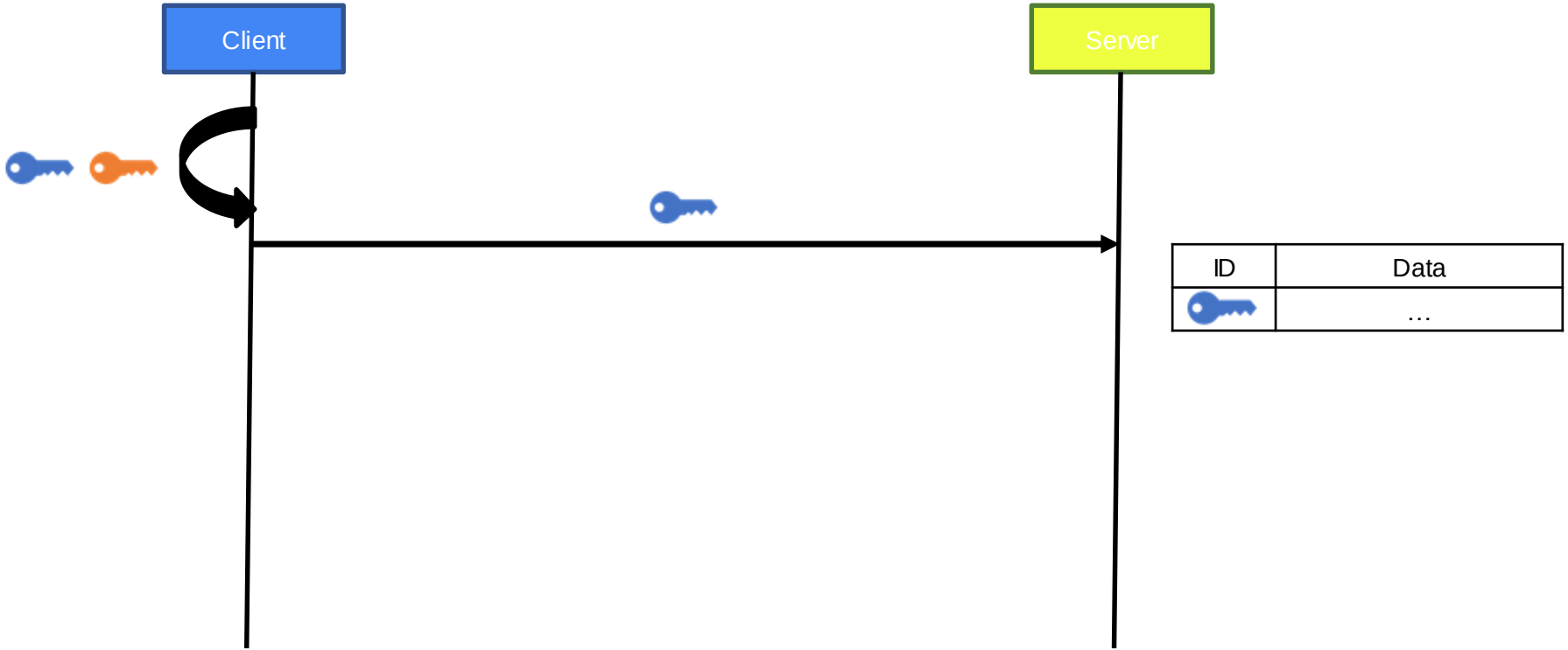
Accountless

- Ad-hoc
- Insecure
- Privacy-invasive

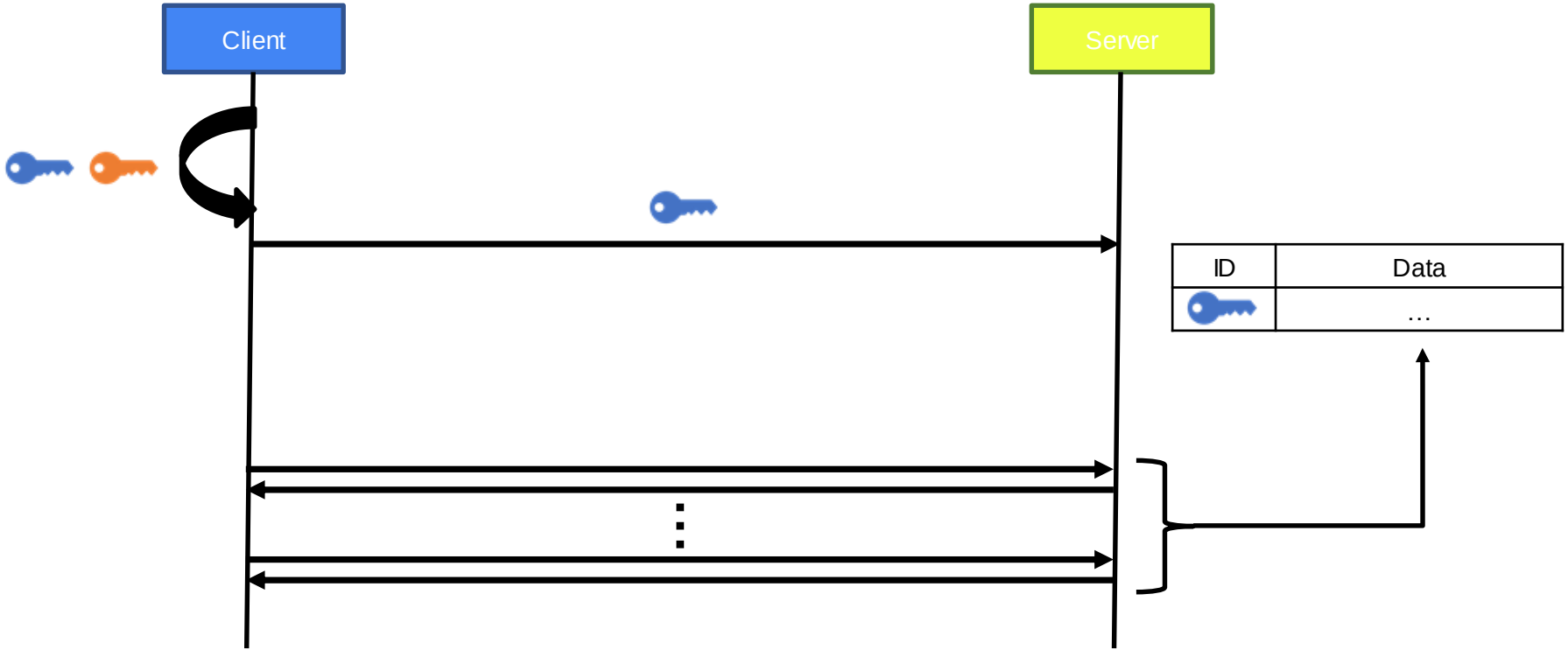
VCR Requirements

- **Unforgeability**
- **Replay resistance**
- **Consumer Privacy**
  - Request – Specific consumer
  - Multiple requests – Single consumer

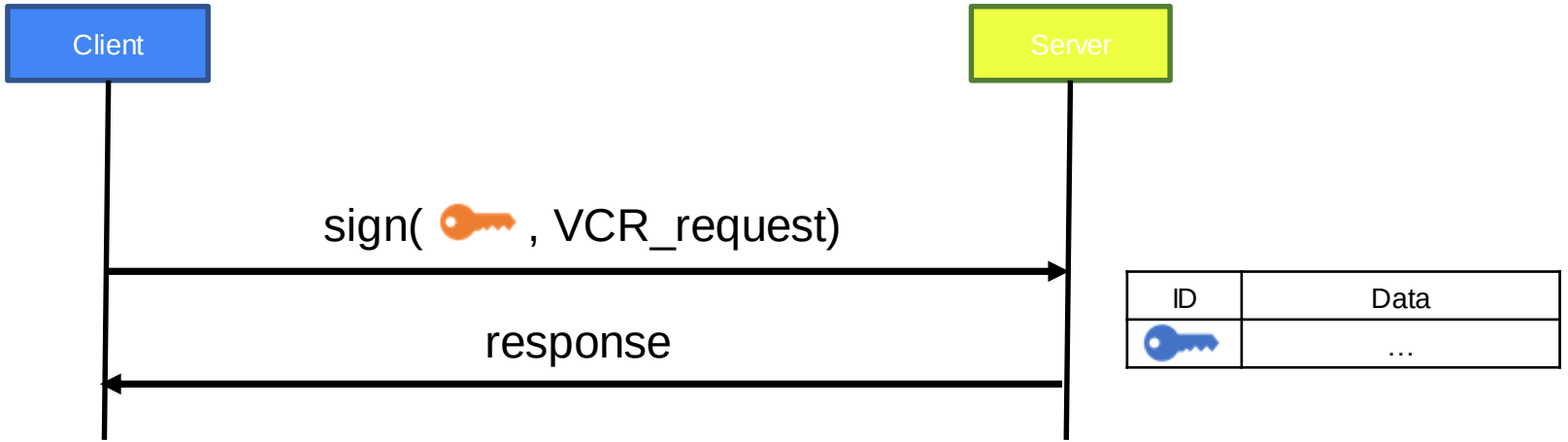
Asymmetric Solution: Setup & Interaction



Asymmetric Solution: Setup & Interaction



Asymmetric Solution: VCR Issuance



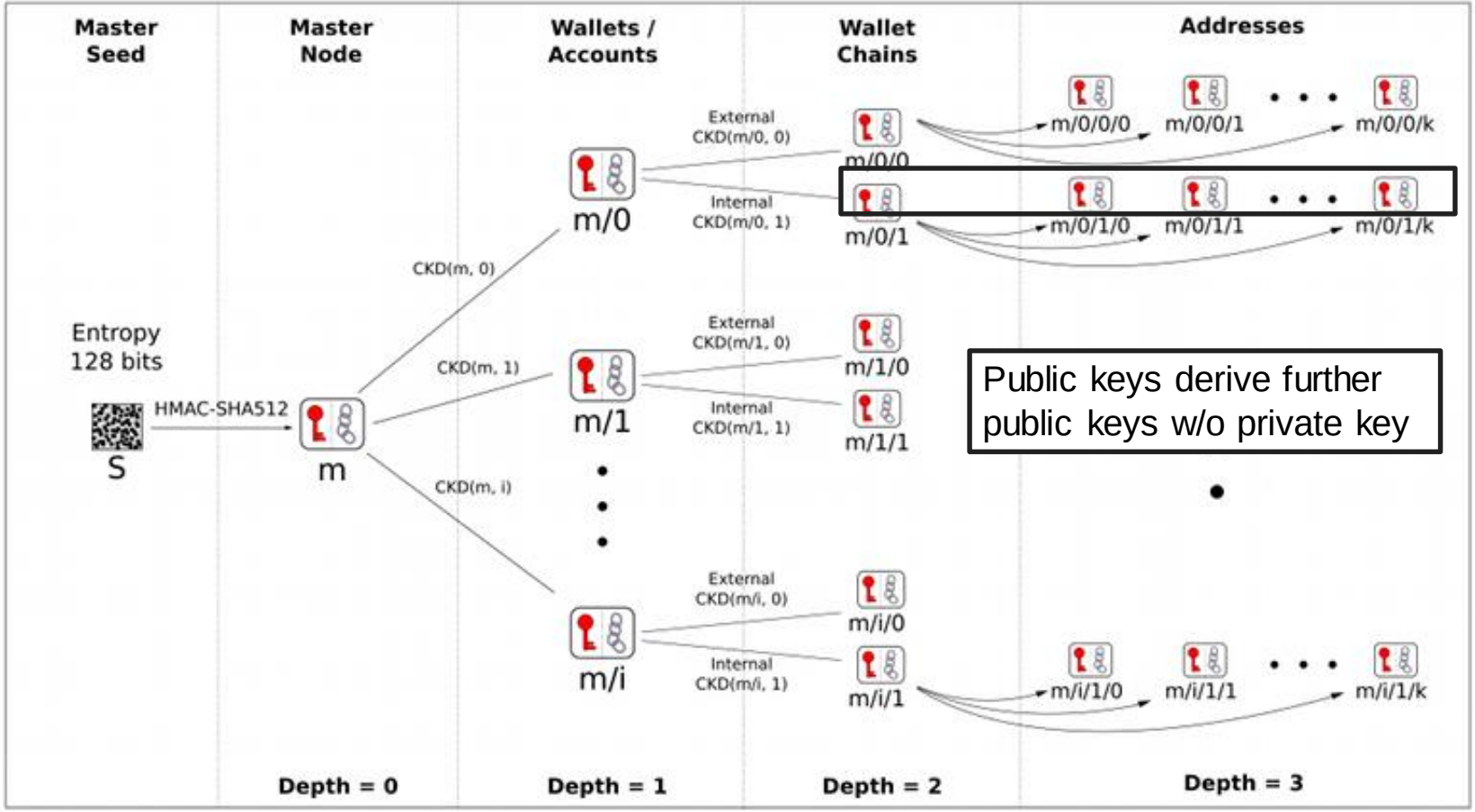
Challenges

- **Key Explosion**
- **Secure Key Management**
- **Long-Term Storage**
- **Multiple Device Support**
- **Server-side storage modification**

VICEROY: GDPR-/CCPA-compliant Verifiable Accountless Consumer Requests

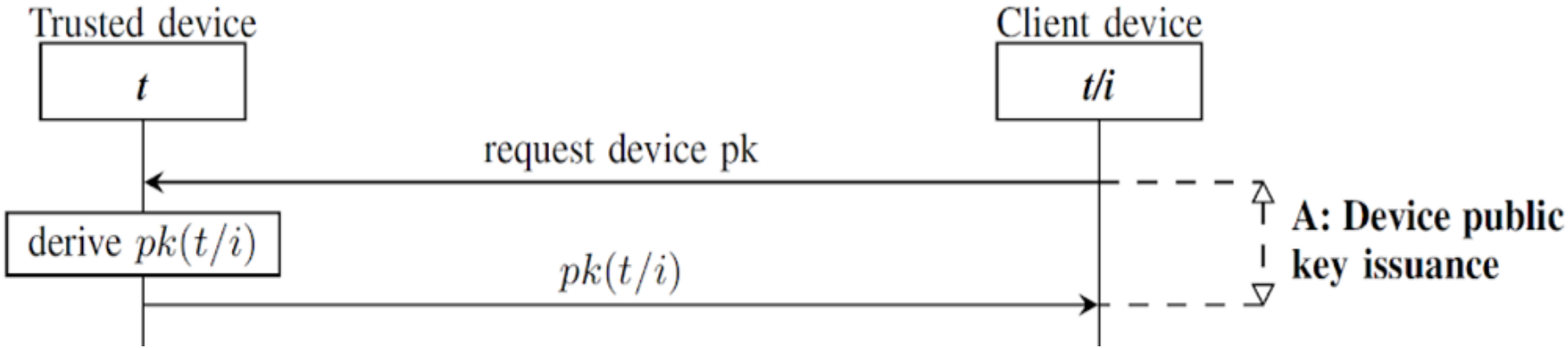
- **Key Explosion**
  - Session-based key generation based on BIP32
- **Secure Key Management:**
  - Trusted consumer device
- **Long-Term Storage:**
  - Untrusted, third-party storage
- **Multiple Device Support:**
  - Device-independent key generation and synced storage
- **Server-side storage modification:**
  - Cookie wrapper

BIP 32 - Hierarchical Deterministic Wallets

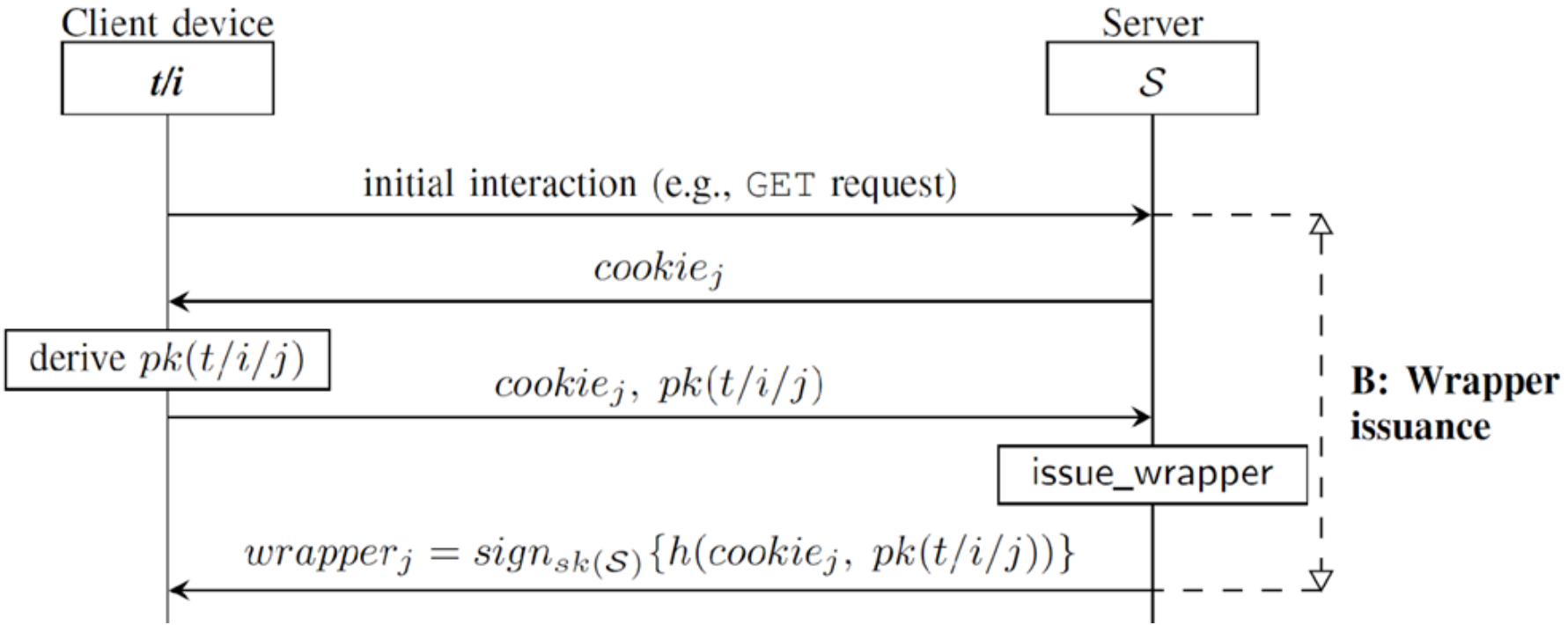


Child Key Derivation Function ~ $CKD(x,n) = HMAC\text{-}SHA512(x_{Chain}, x_{PubKey} || n)$

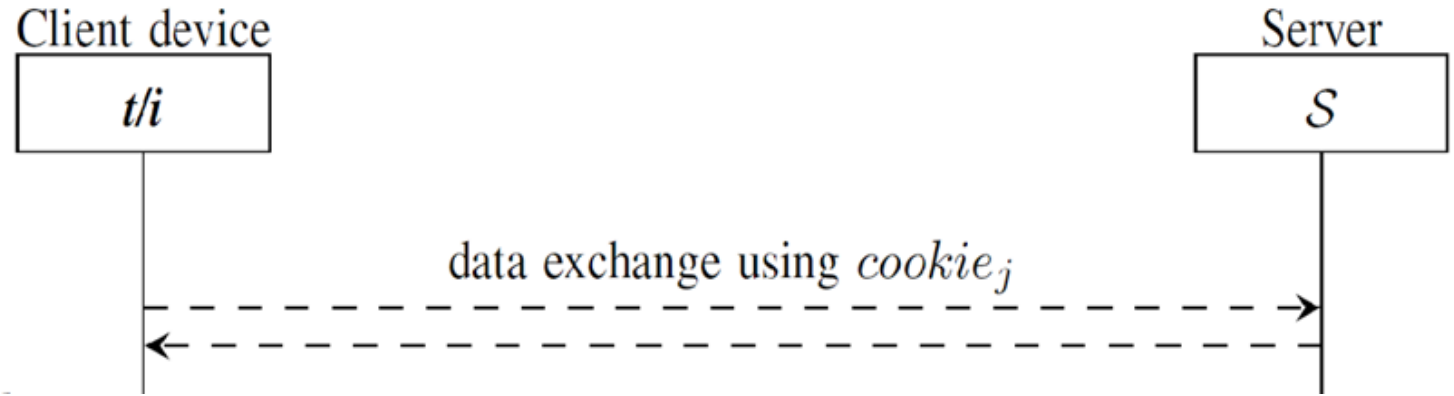
VICEROY - Device provisioning



VICEROY - Wrapper issuance



VICEROY - Data exchange



VICEROY - VCR Issuance