Integrity and Metadata Protection in Data Retrieval

Kirill Nikitin

Decentralized and Distributed Systems Laboratory

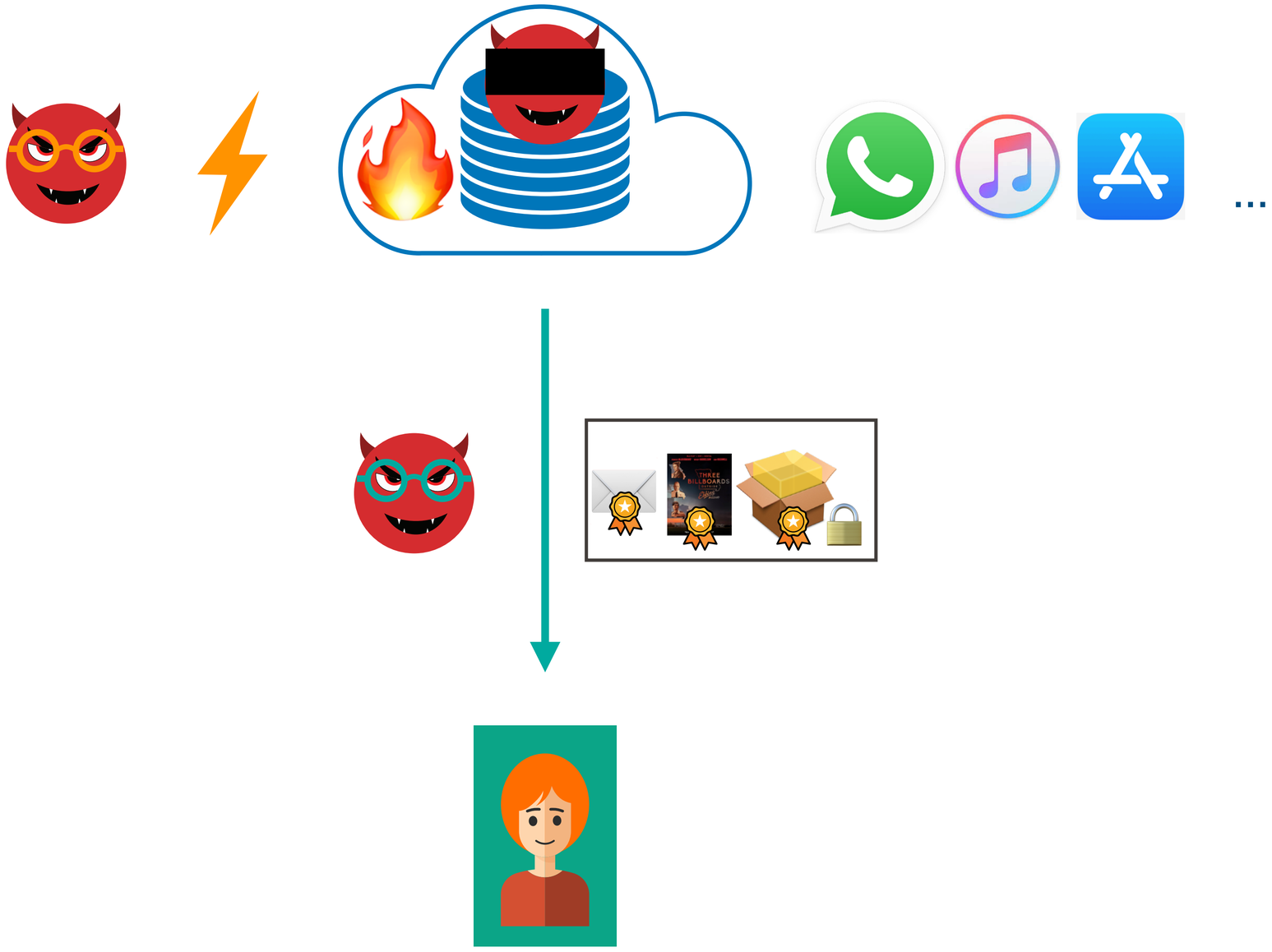
Public PhD defense, 26.11.2021

Thesis Director:	Prof. Bryan Ford
Jury President:	Prof. Jean-Pierre Hubaux
Examiners:	Prof. Katerina Argyraki
	Prof. Justin Cappos
	Prof. Srdjan Capkun

Data retrieval all day long

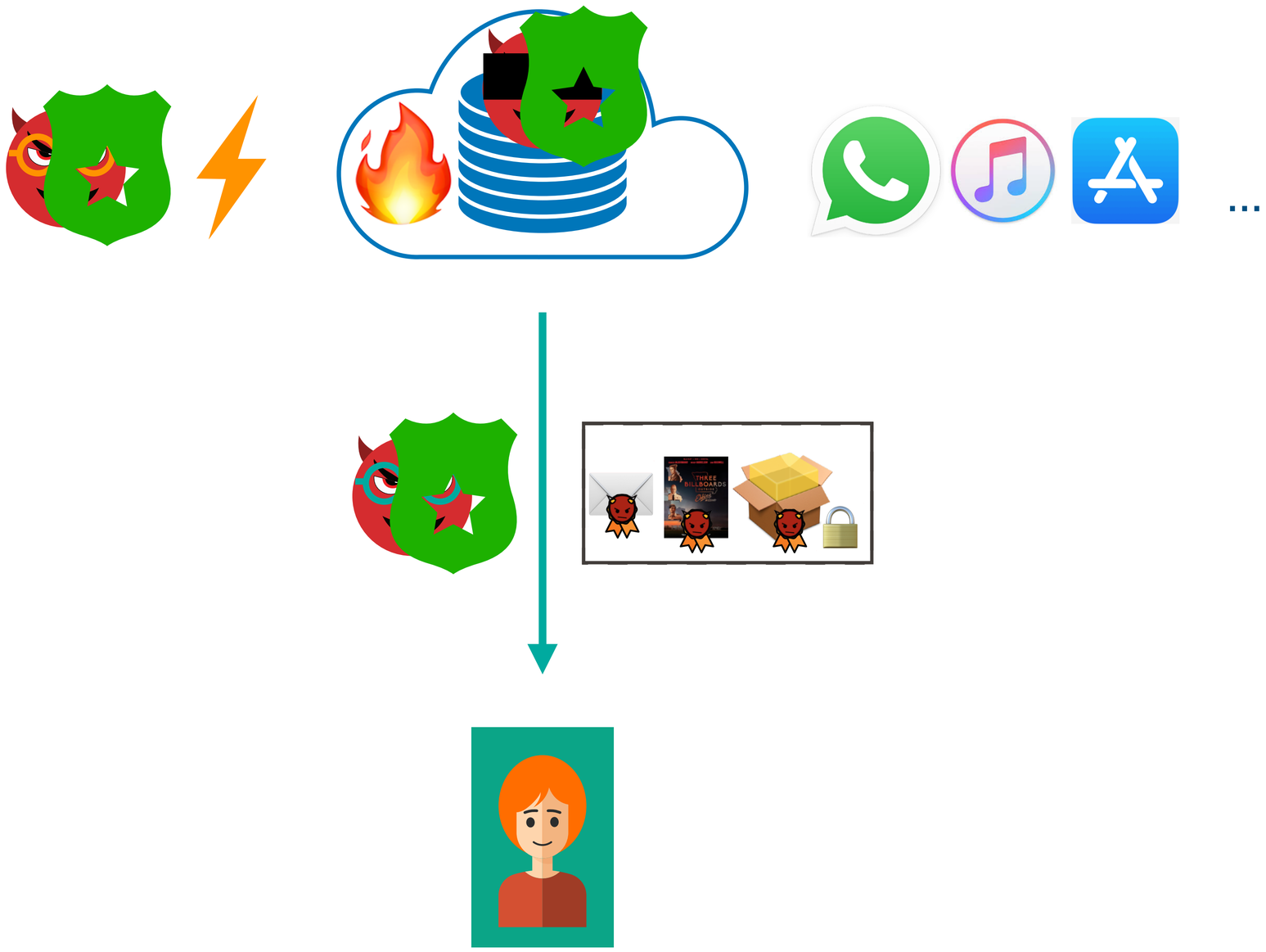
Image from Unsplash

Data retrieval all day long



Credit for the user and evil pictures here and graphics afterwards is to vecteezy.com

Current protection mechanisms do not suffice



Credit for the user and evil pictures here and graphics afterwards is to vecteezy.com

This thesis

On-the-network attacker

- Protecting encryption metadata (Chapter 2) [1]

Malicious provider

- Data integrity in single-server private information retrieval (Chapter 3) [2]

Compromised provider

- Securing retrieval of software updates (Chapter 4) [3]

[1] K. Nikitin, L. Barman, W. Lueks, M. Underwood, J.-P. Hubaux, and B. Ford, “Reducing Metadata Leakage from Encrypted Files and Communication with PURBs”, PETS 2019.

[2] S. Colombo, K. Nikitin, B. Ford, and H.Corrigan-Gibbs, “Authenticated Private Information Retrieval”, Under submission.

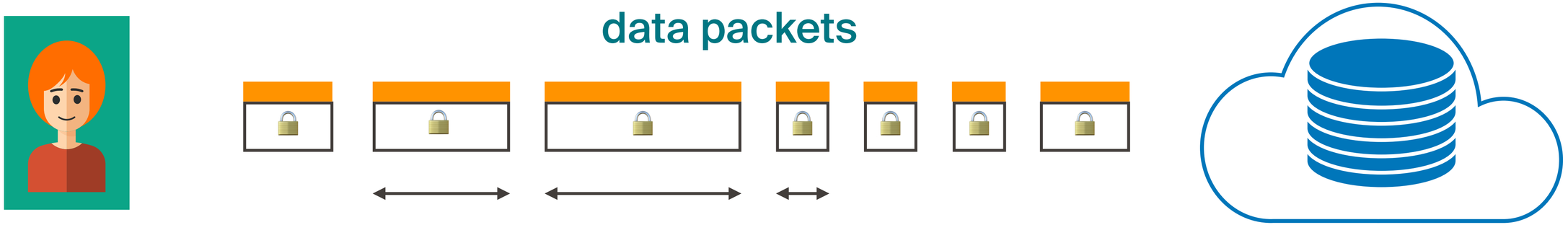
[3] K. Nikitin, E. Kokoris-Kogias, P. Jovanovic, N. Gailly, L. Gasser, I. Khoffi, J. Cappos, and B. Ford, “CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds”, USENIX Security 2017.

What are metadata and integrity?

Metadata are data about data

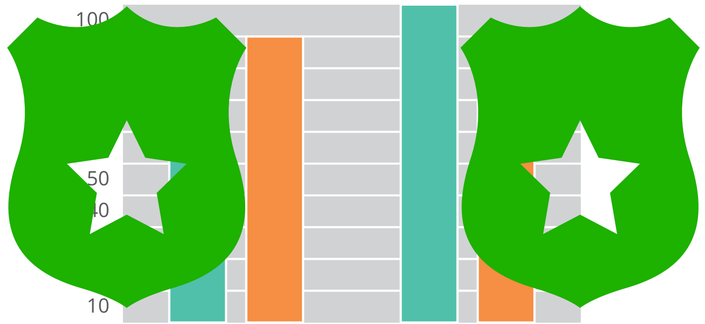
Image from Unsplash

Metadata are data about data



Data integrity and integrity of a system

- Data integrity is the assurance of data consistency and accuracy



- A computer system’s integrity is the ability to withstand compromise despite its weaknesses

One of the truest tests of integrity is its blunt refusal to be compromised

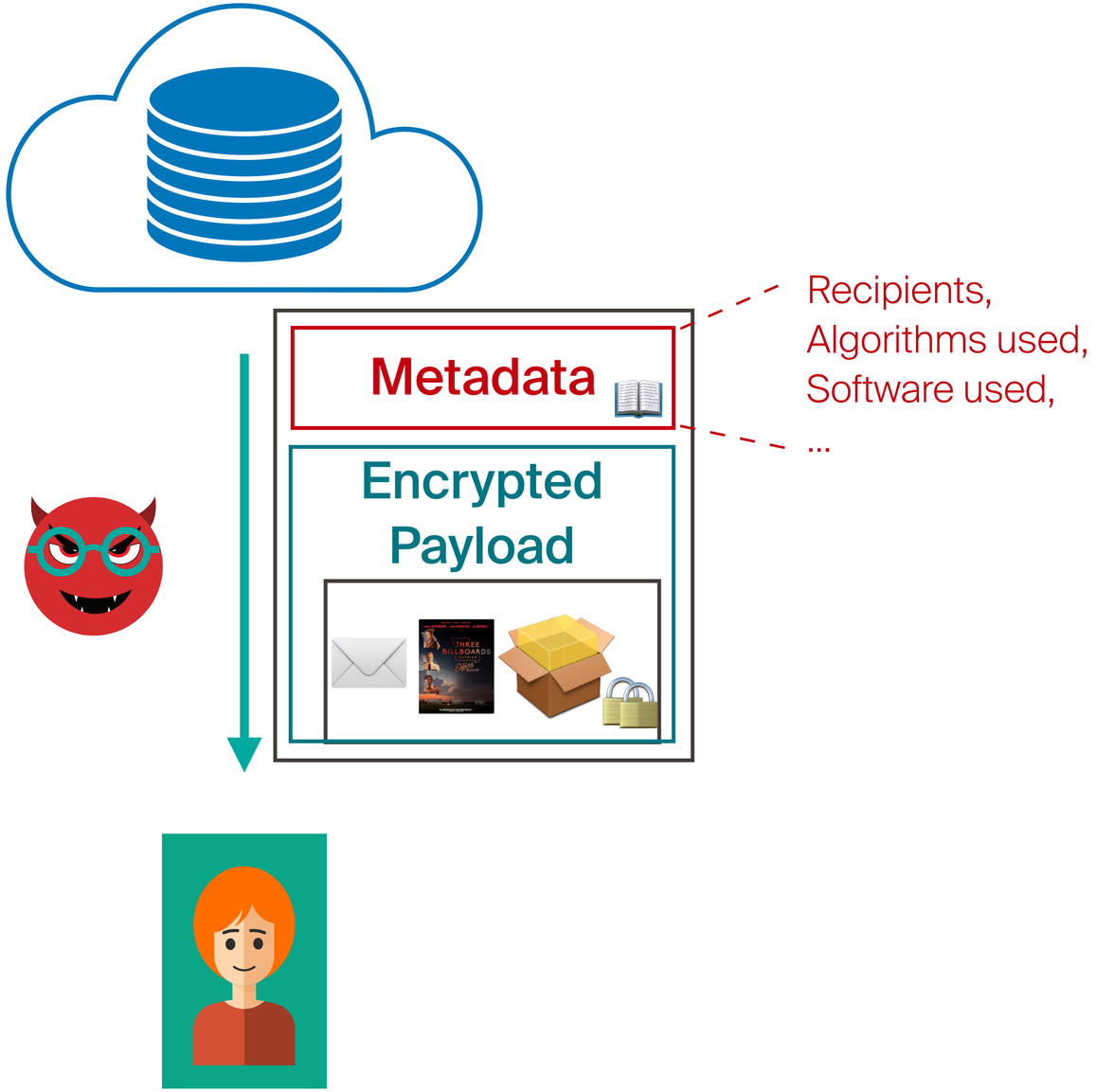
Roadmap

- Introduction
- Protecting encryption metadata (Chapter 2)
- Data integrity in single-server PIR (Chapter 3)
- Securing retrieval of software updates (Chapter 4)
- Conclusion

Roadmap

- Introduction
- Protecting encryption metadata (Chapter 2)
- Data integrity in single-server PIR (Chapter 3)
- Securing retrieval of software updates (Chapter 4)
- Conclusion

Metadata exposure in ciphertexts



OpenPGP Packet Format

```
8502 0c01 9497 608d d051 8f79 010f ff46
bd7f 1821 27a9 42c4 01b4 7ecd 433e 7f90
. . . . . 74b8 139c a802 6678 ba0d 1abd
-------------------------------------.
d264 014b 6a5a f586 e3fa b98e 92d1 6759
7186 2ccc ac50 3db7 fa03 4f31 dcd7 fa40
. . . . . 4b09 d9b4 1654 972d 5c22 47db
```

OpenPGP Packet Format

Packet Type

Format version

Recipient Key ID

Public-Key Algorithm

```
8502 0c01 9497 608d d051 8f79 010f ff46
bd7f 1821 27a9 42c4 01b4 7ecd 433e 7f90
. . . . . 74b8 139c a802 6678 ba0d 1abd
-----------------------------------------
d264 014b 6a5a f586 e3fa b98e 92d1 6759
7186 2ccc ac50 3db7 fa03 4f31 dcd7 fa40
. . . . . 4b09 d9b4 1654 972d 5c22 47db
```

Session Key Part

Data Part

Encrypted Data →

Is exposing encryption metadata necessary?

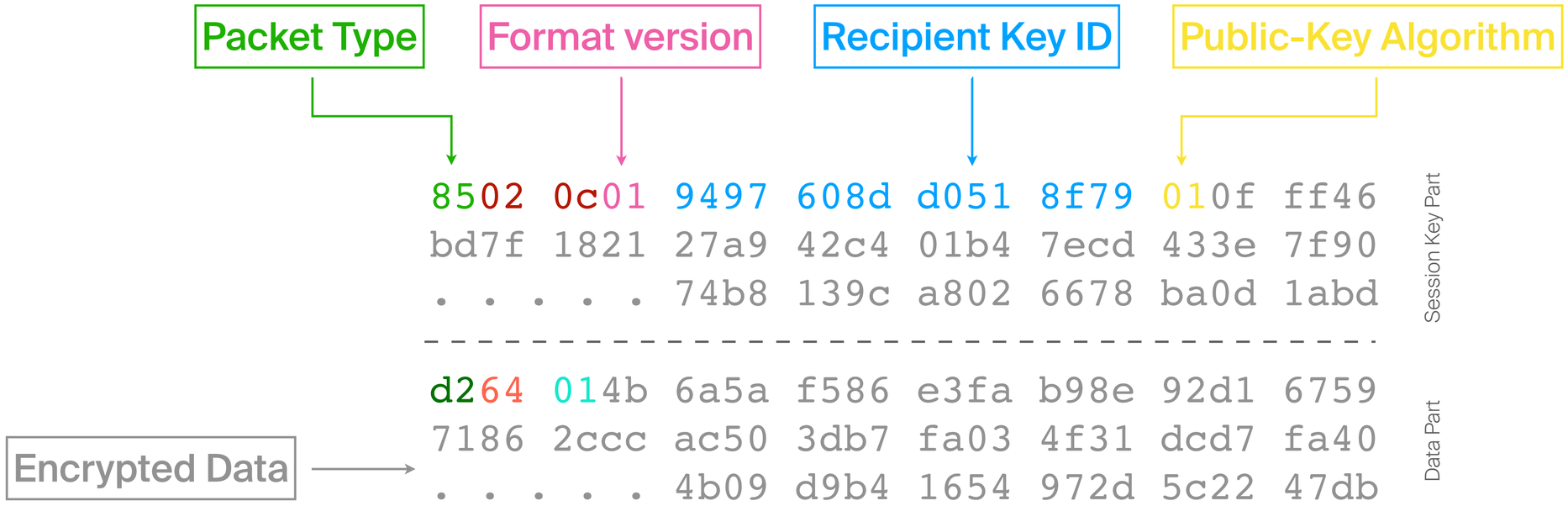
An OpenPGP message to Martin Vetterli encrypted with RSA-512 using an outdated format??

Small key? Outdated format? I might crack it!

Avoiding metadata leakage

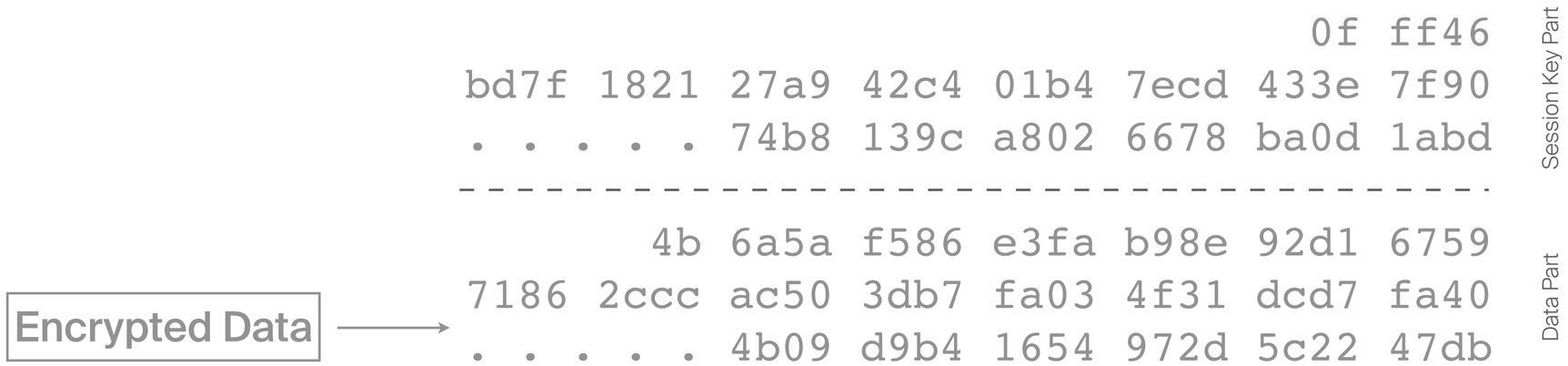
- Can we design an application-level ciphertext format that avoids leakage of encryption metadata?
- Encryption metadata concretely:
  - The ciphertext's intended recipients
  - The encryption algorithm used
  - What application has produced the ciphertext
  - ...

What If We Stripped Off All the Metadata?



What If We Stripped Off All the Metadata?

Encrypt the metadata instead!



- How does a recipient parse a ciphertext without any auxiliary information?
- What if the ciphertext is encrypted
  - To multiple recipients
  - By using multiple cryptographic algorithms

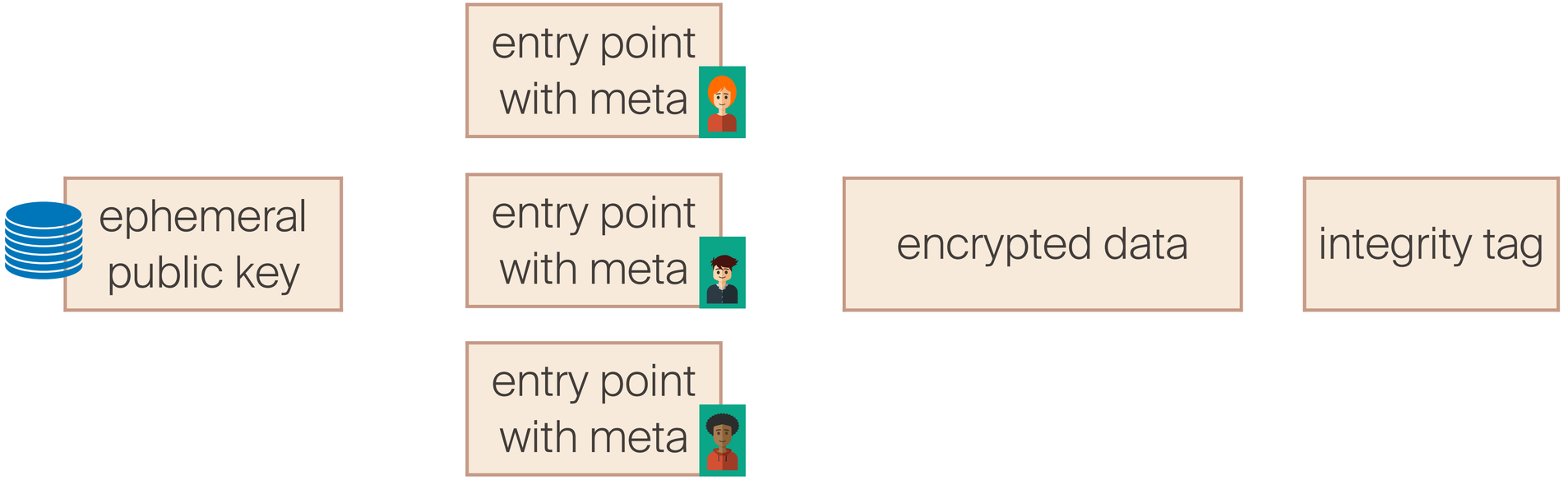
Padded Uniform Random Blobs (PURBs)

- A ciphertext format for application data without *any* metadata in clear
- Generic, i.e., still works efficiently with a large number of recipients and encryption algorithms used
- A PURB must be indistinguishable from a random bit string

PURBs encoding insights

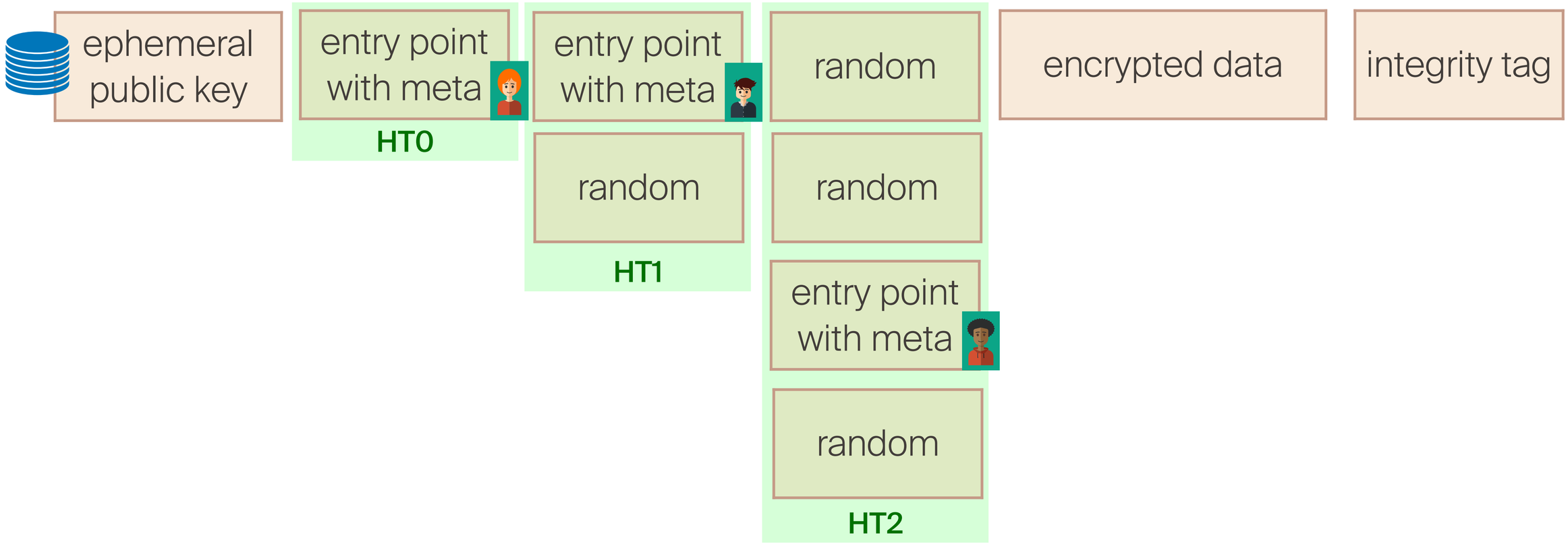
The metadata can be found efficiently by trial decryptions following a predefined logic

✓ Data deduplication and cryptographic agility via layering



PURBs encoding insights

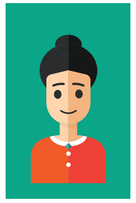
✓ Efficient decryption without cleartext metadata via structure definition



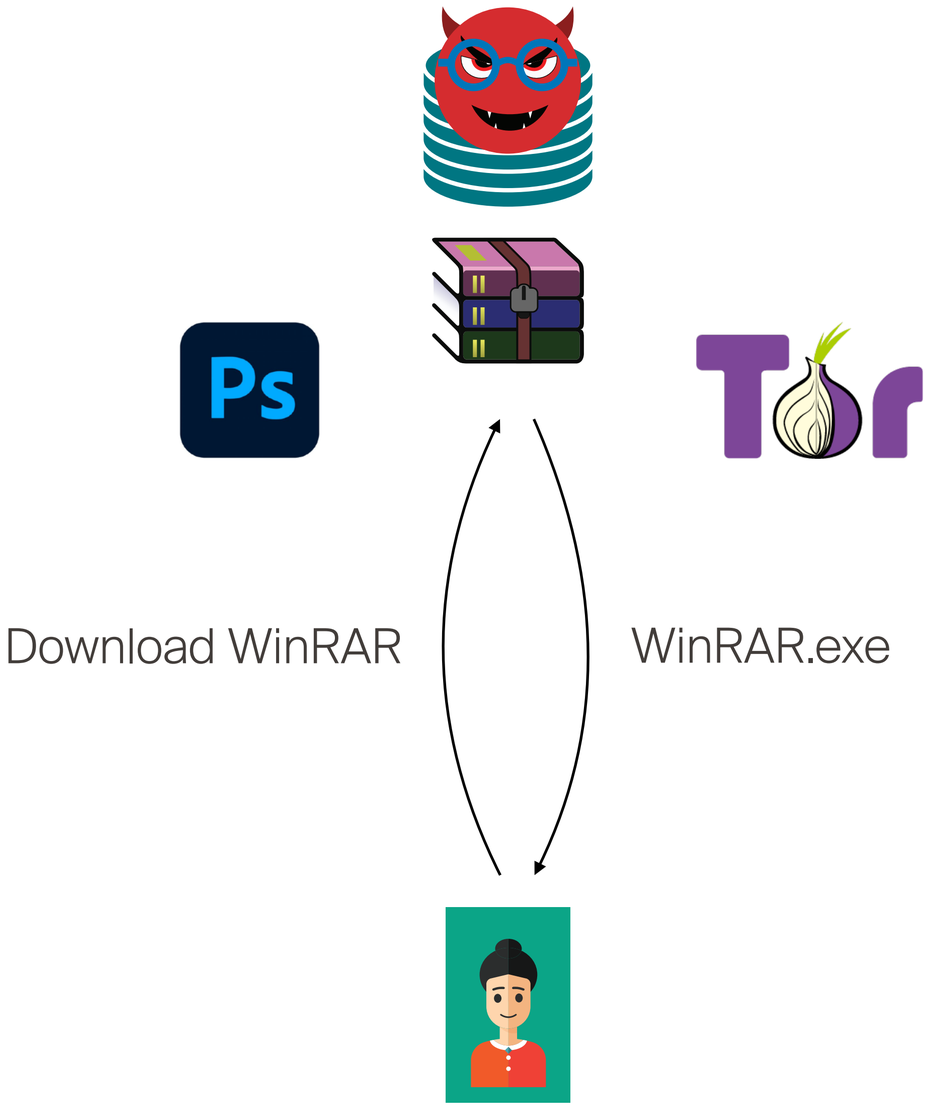
Roadmap

- Introduction
- Protecting encryption metadata (Chapter 2)
- Data integrity in single-server PIR (Chapter 3)
- Securing retrieval of software updates (Chapter 4)
- Conclusion

Service providers learn user's choices

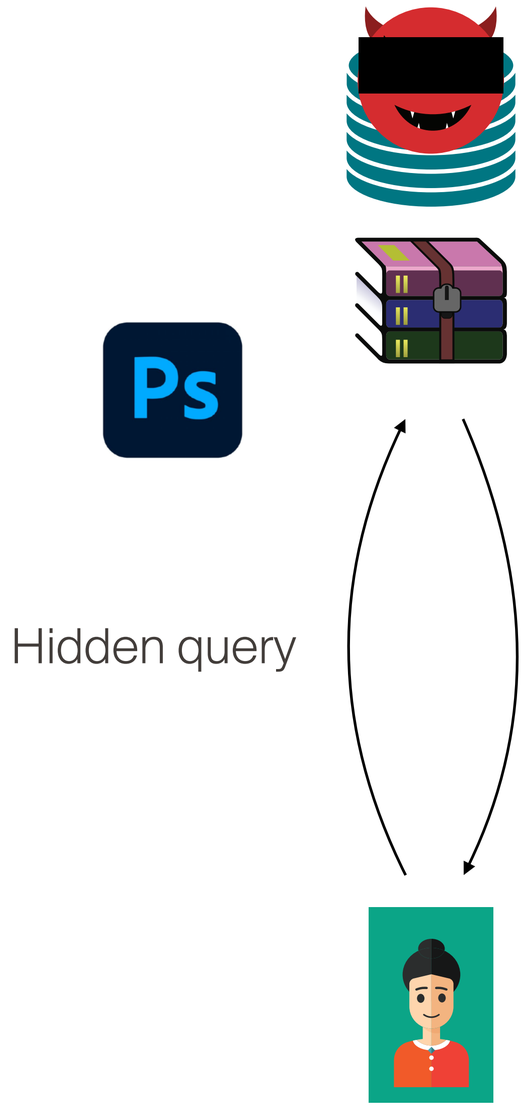


Service providers learn user’s choices



Only 20’000$!

Private Information Retrieval (PIR)

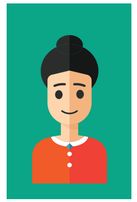


Blind computation

Some applications:
software updates [Cap13]
online-presence service [BDG15]
anonymous messaging [AS16]
video streaming [GCM+16]
encrypted search [DFL+20]

The single-server PIR setting

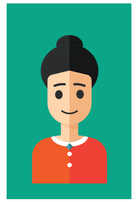
Problem: No data integrity by default



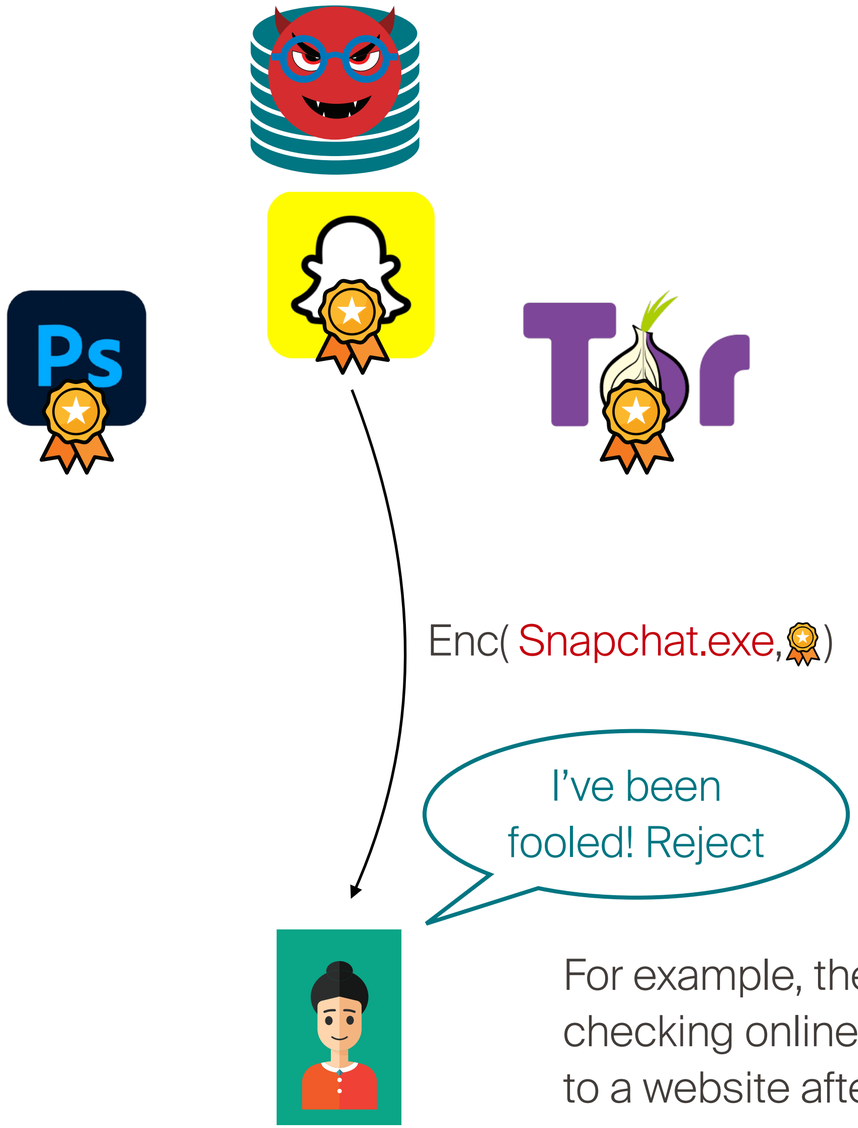
A typical way to get integrity

Attach a digital signature to each record!

Signing data owner

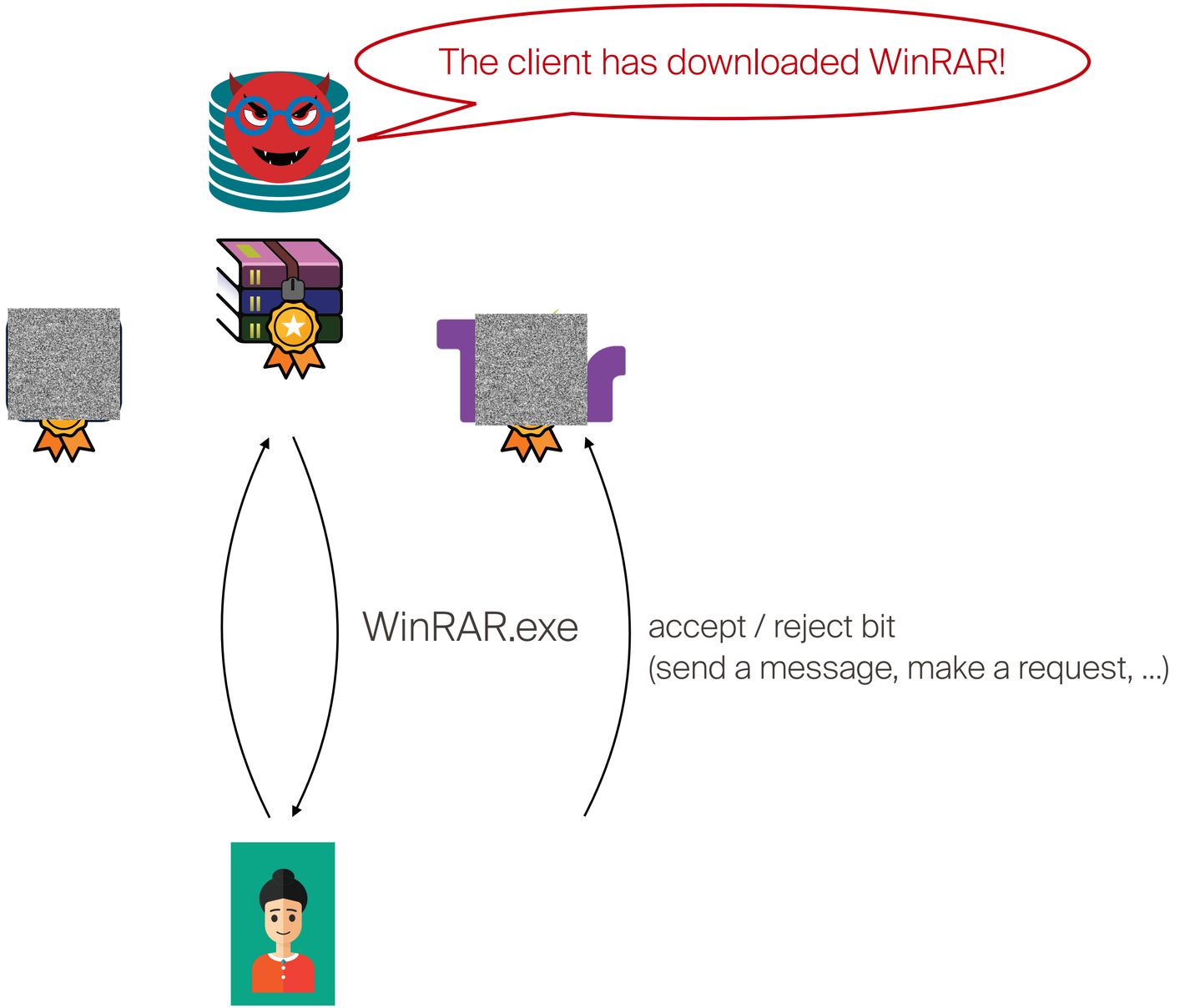


When integrity breaks privacy



For example, the client starts communication after checking online presence of a friend, or connects to a website after retrieving a DNS record, etc

When integrity breaks privacy



Verifiable single-server PIR

- Provides privacy *and* integrity atomically
- Client detects any altering of the database, even for the records she is *not* retrieving

Verifiable single-server PIR

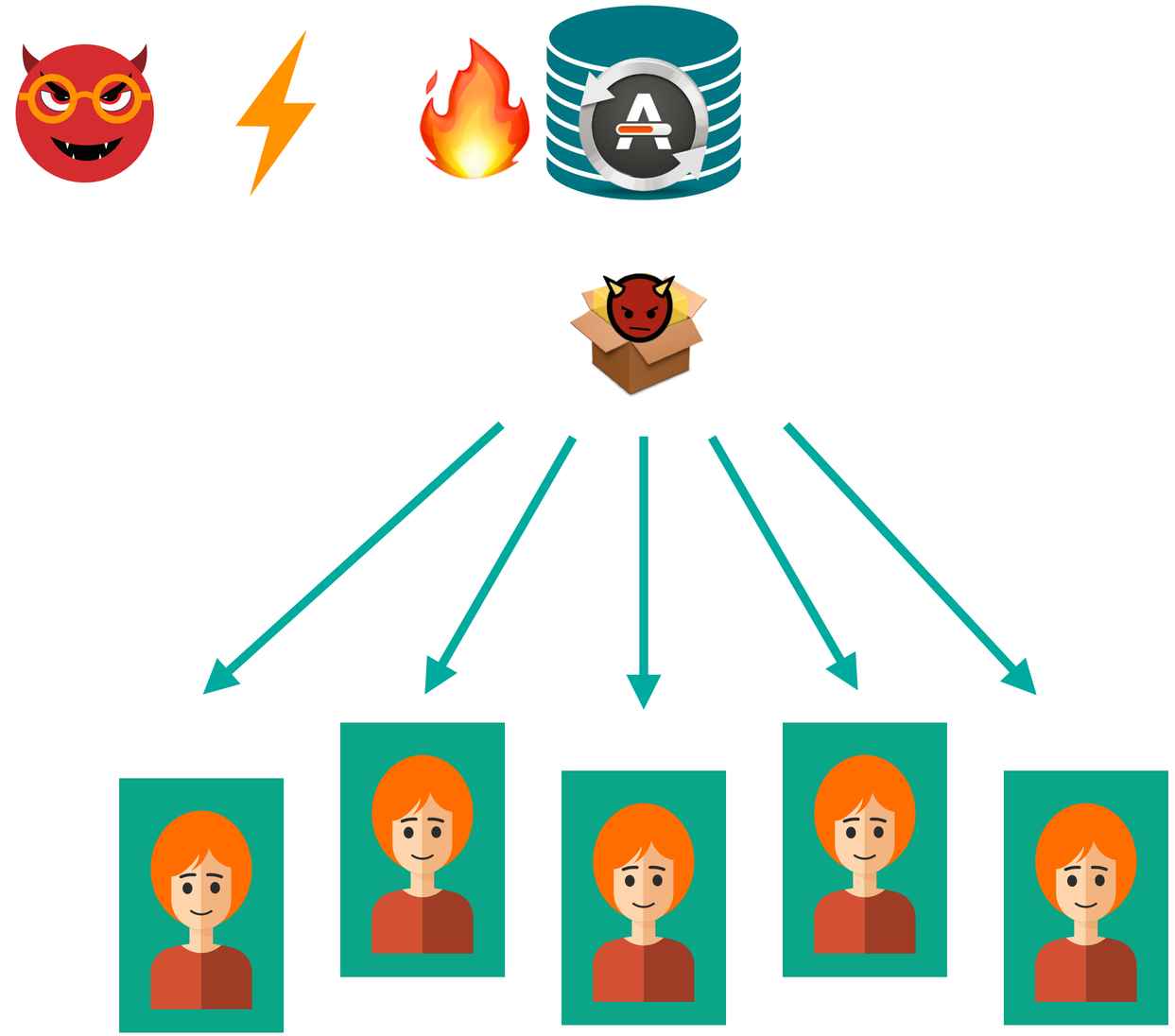
Public digest committing the server to the database content

Client sends a randomized query to the server and validates that the digest is present in the server’s response after de-randomization

Roadmap

- Introduction
- Protecting encryption metadata (Chapter 2)
- Data integrity in single-server PIR (Chapter 3)
- Securing retrieval of software updates (Chapter 4)
- Conclusion

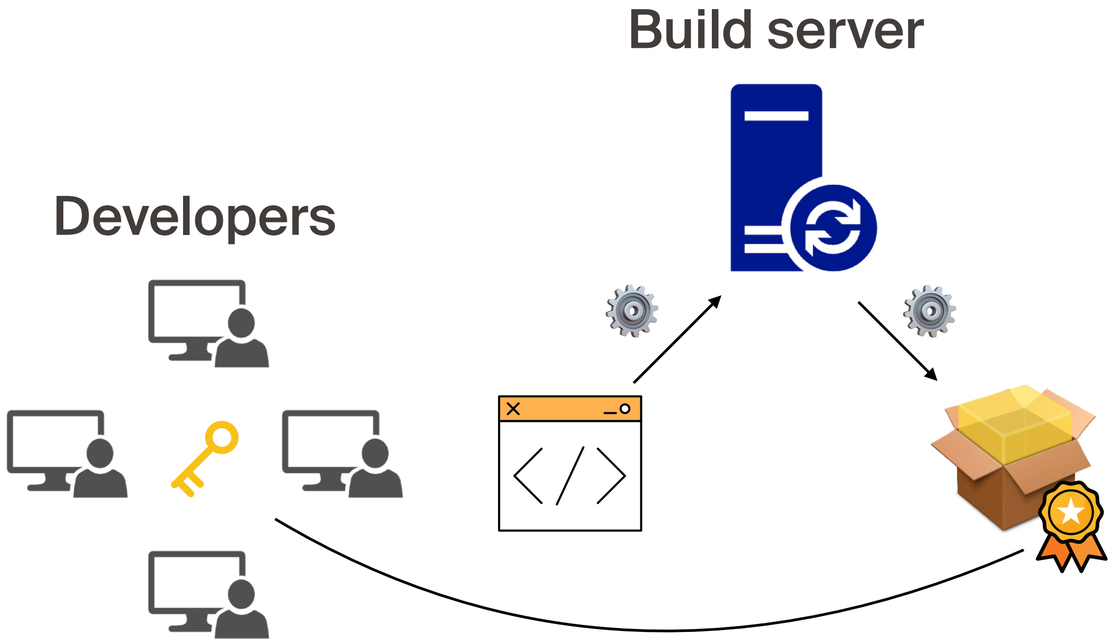
Compromising a software-update system

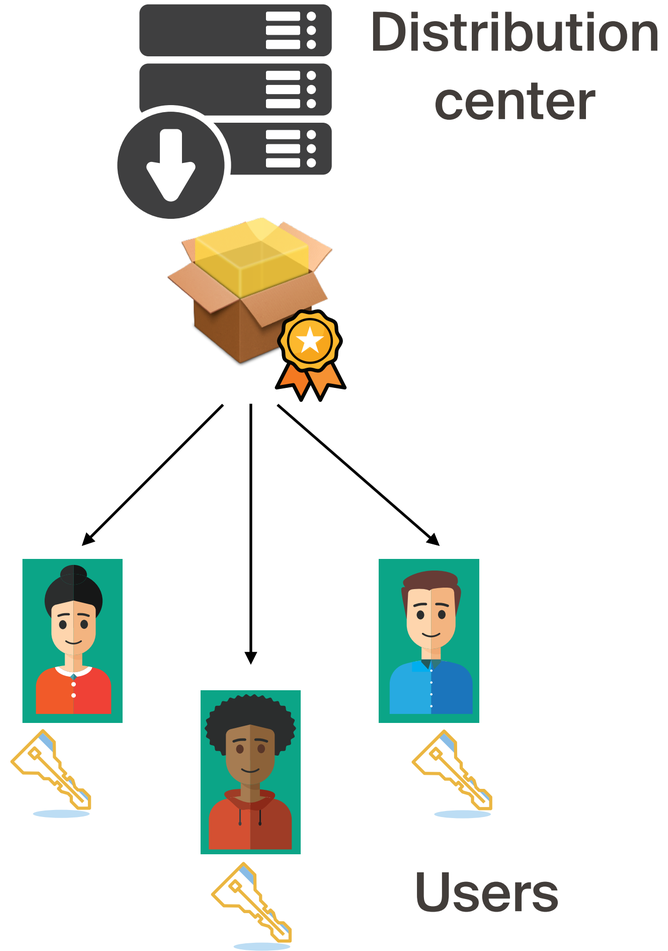


Compromised software-update systems

Software Release Pipeline

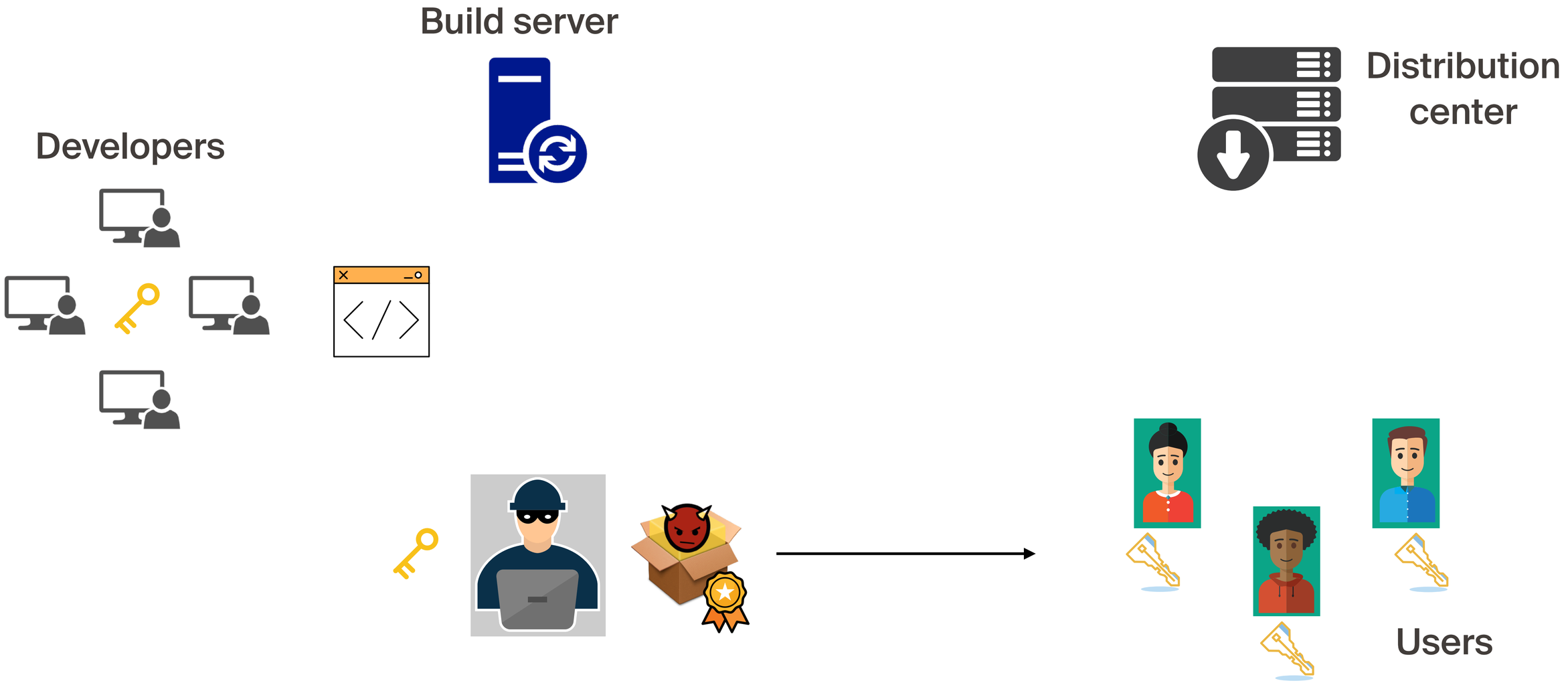
Development/Review → Building release binaries → Sign-off → Release distribution





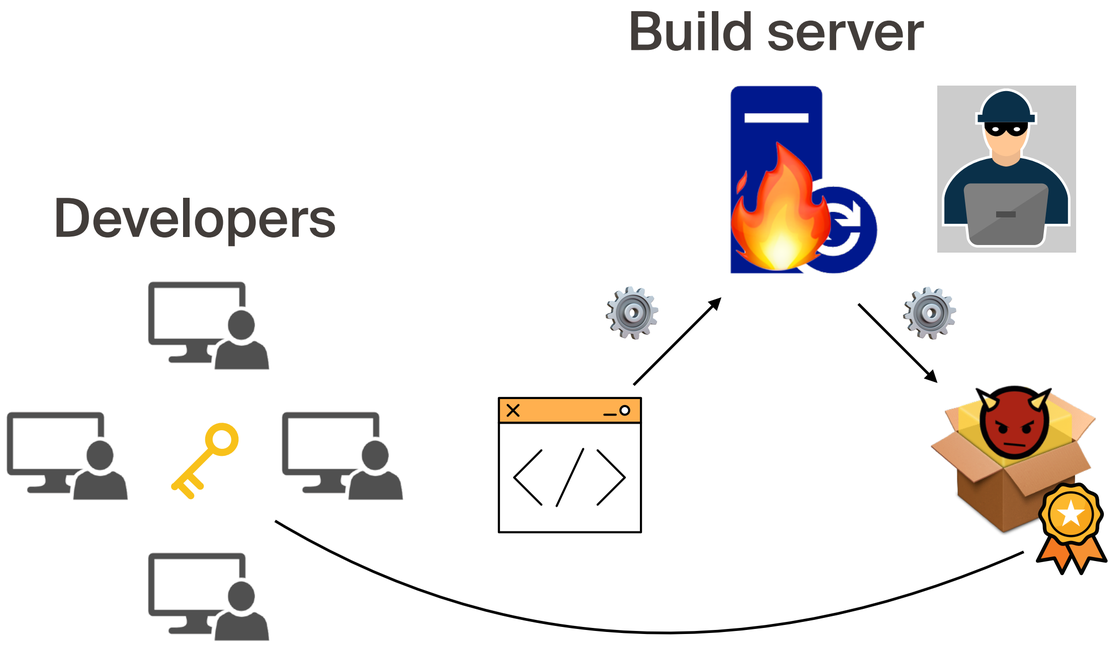
Challenges

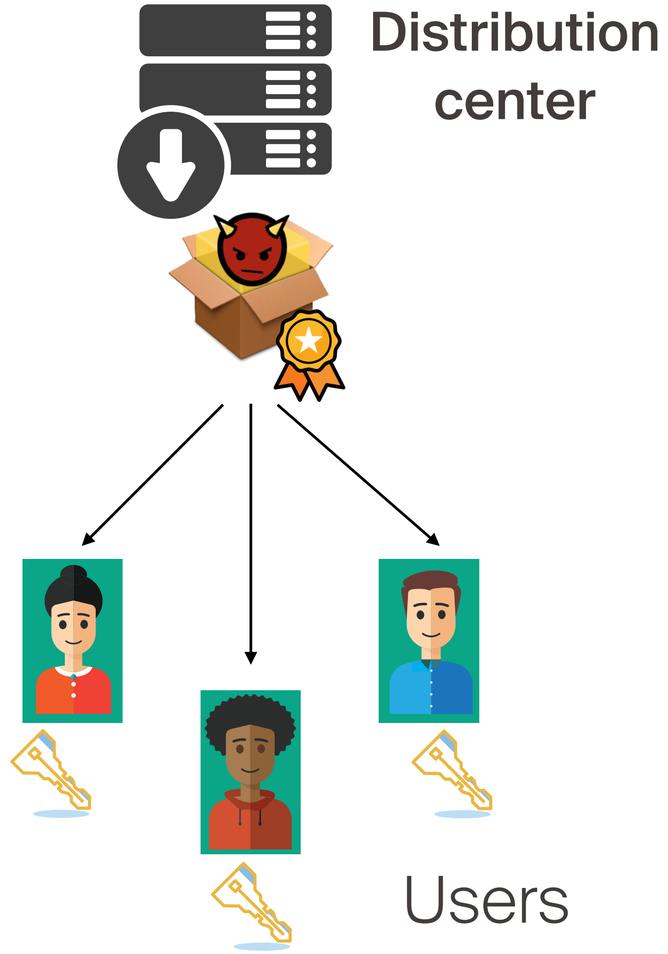
(1) Make software-update process resilient to partial key compromise



Challenges

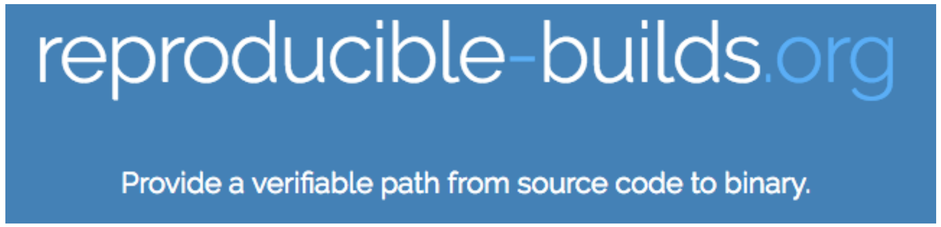
(2) Prevent malicious substitution of a release binary during a build process





Challenges

(2) Prevent malicious substitution of a release binary during a build process

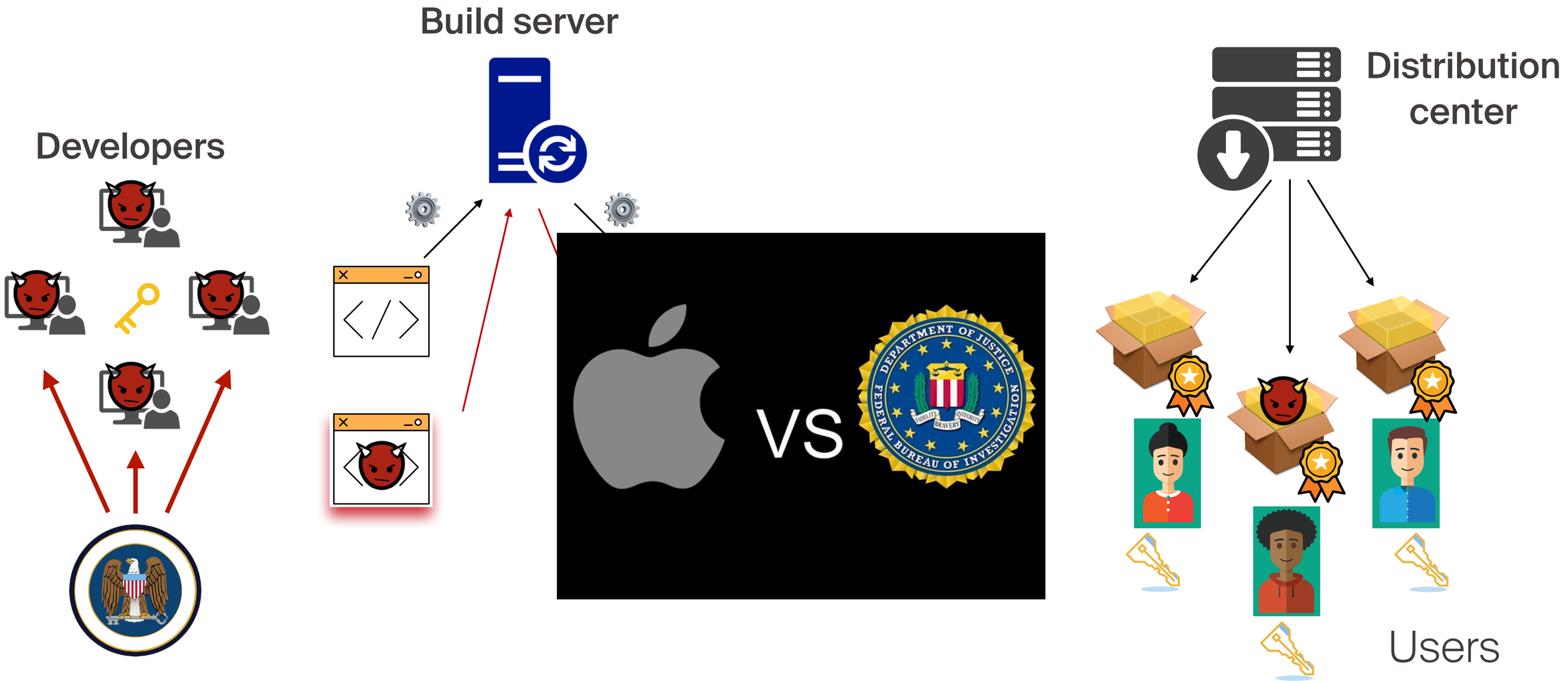


Over 90% of the source packages included in Debian 9 will build bit-for-bit identical binary packages

1. **Regular users do not compile from source code**
2. **Reproducible compilation can take hours (e.g., Tor browser)**

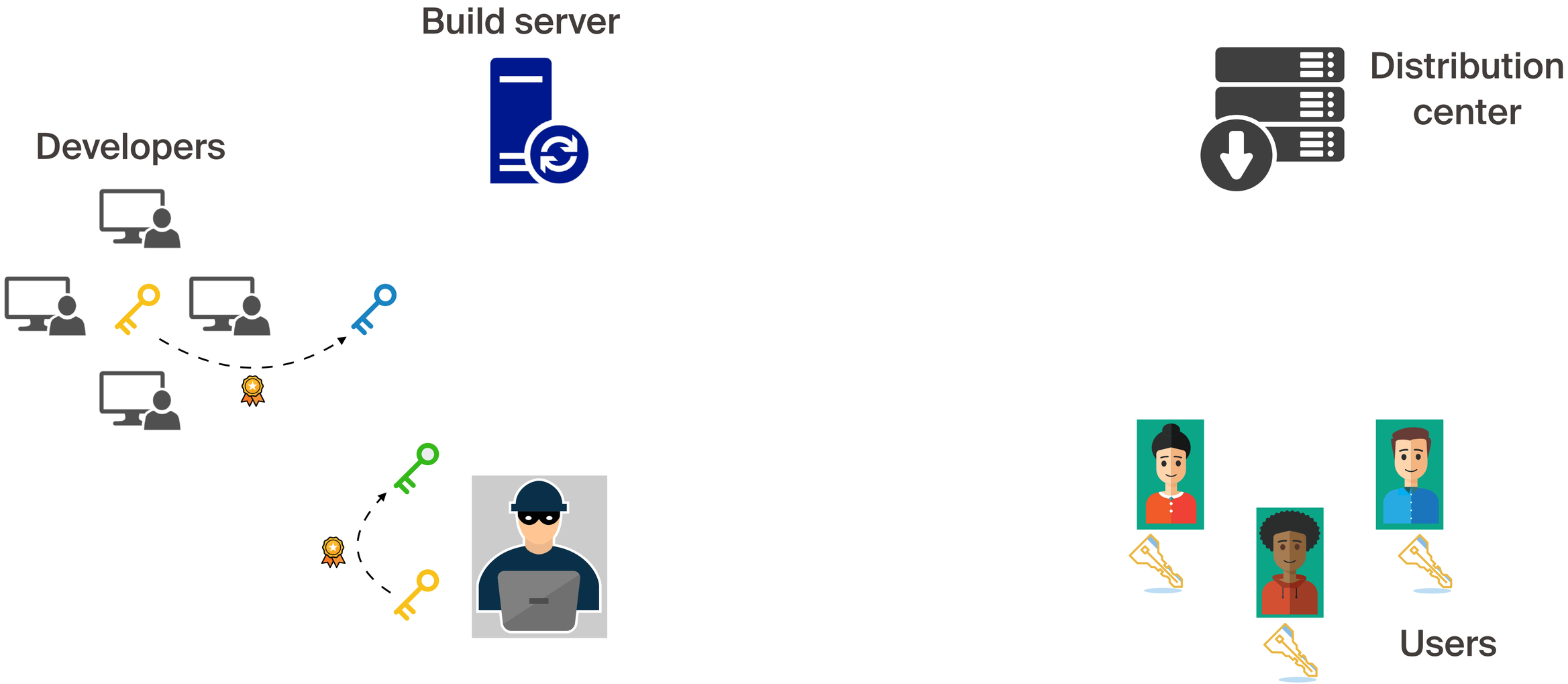
Challenges

(3) Protect users from targeted attacks by coerced or bribed developers



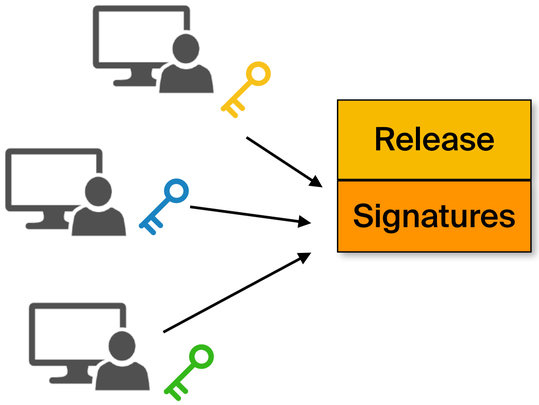
Challenges

(4) Enable developers to securely rotate their signing keys in case of renewal or compromise

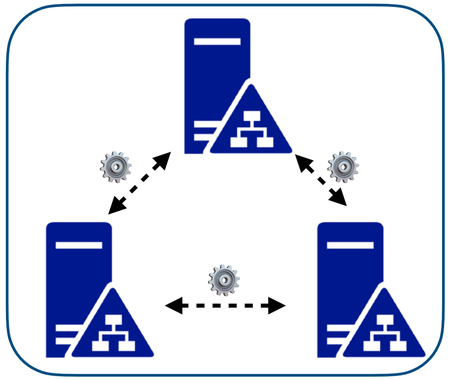


CHAINIAC: Securing software-update retrieval

Decentralized release approval

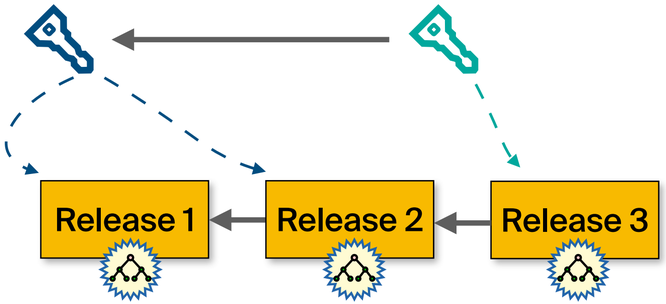


Third-party release witnessing and validation



Anti-equivocation via transparency release log

Key evolution via additional key history log



Roadmap

- Introduction
- Protecting encryption metadata (Chapter 2)
- Data integrity in single-server PIR (Chapter 3)
- Securing retrieval of software updates (Chapter 4)
- Conclusion

Conclusion

- Protecting complex processes requires comprehensive solutions
- Neglecting seemingly irrelevant security properties can lead to subtle vulnerabilities and design flaws
- Hybrid mechanisms that provably provide multiple security properties in an atomic way is the right direction forward

Conclusion

On-the-network attacker

- Protecting encryption metadata

Malicious provider

- Data integrity in single-server private information retrieval

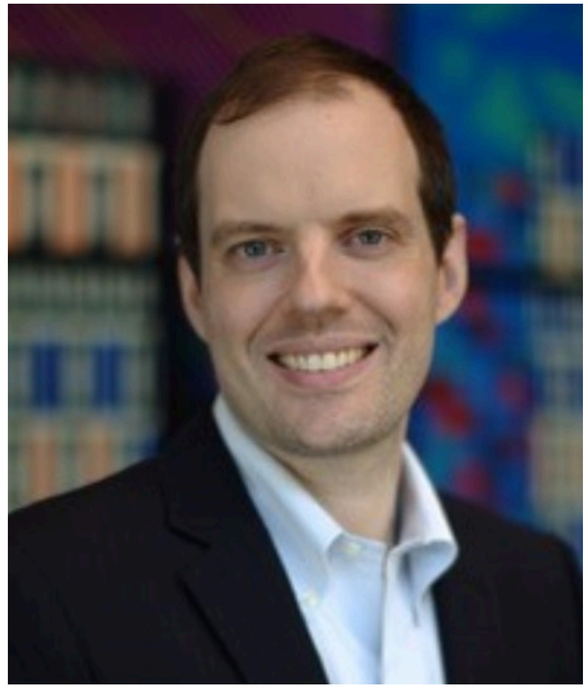
Compromised provider

- Securing retrieval of software updates

Questions?

Acknowledgements

Thesis committee



Srinath Setty

Henry Corrigan-Gibbs

Arjen Lenstra

DEDIS

Cey and Lefteris

The LDS selection

The French club

The Cheese factory

Leo and Dusan

Iegor and Sarah

Parents