# HopAuth: Hop-by-Hop Authentication for Secure In-Network Data Retrieval

Ruidong Li, Hitoshi Asaeda

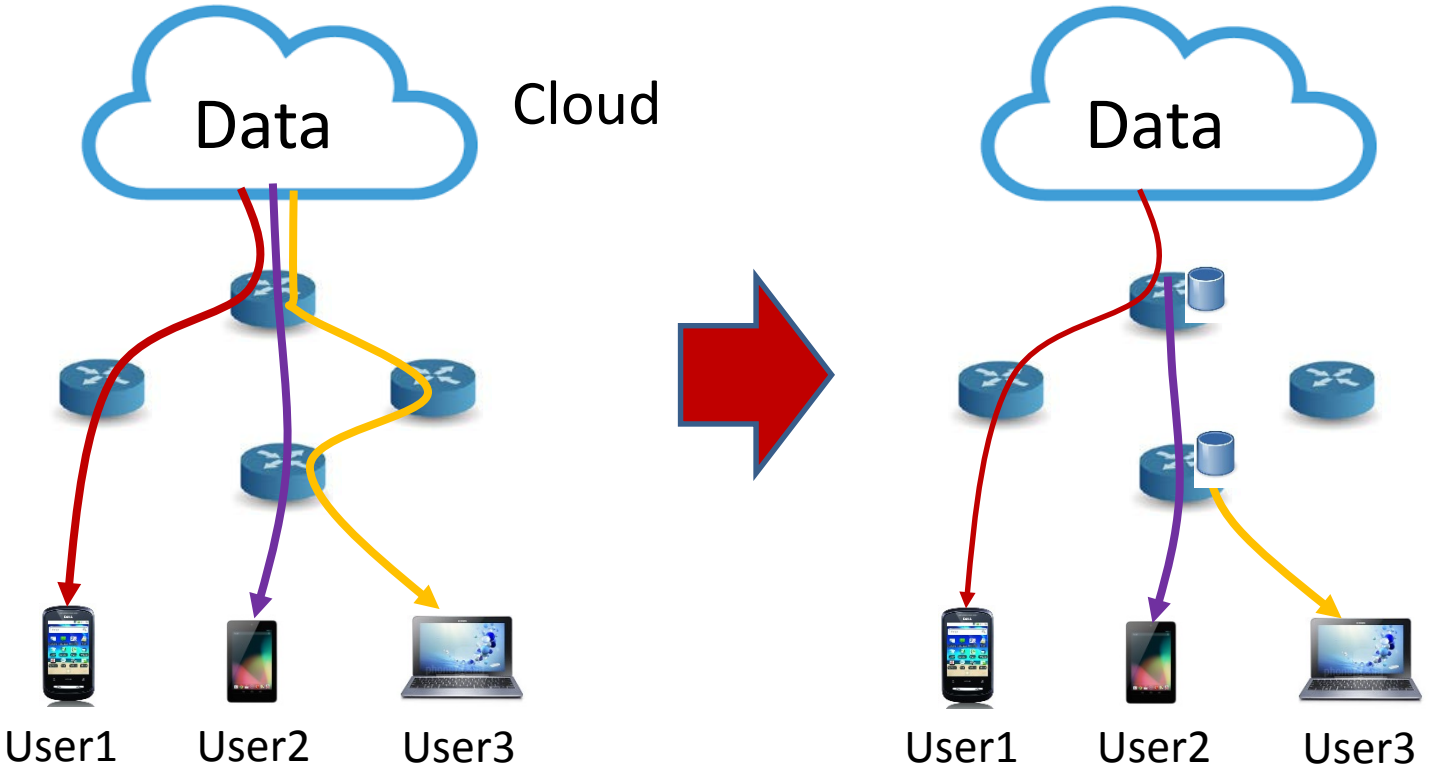
National Institute of Information and Communications Technology (NICT)

"DCAuth: Data-Centric Authentication for Secure In-Network Big-Data Retrieval", IEEE Transactions on Network Science and Engineering (TNSE), Sep. 2018.

# Outline

- In-Network Data Retrieval & Content-Centric Network
- Adversary Model
- Traditional Approach
- Proposed Hop-by-Hop Authentication Mechanism (HopAuth)
- Conclusions

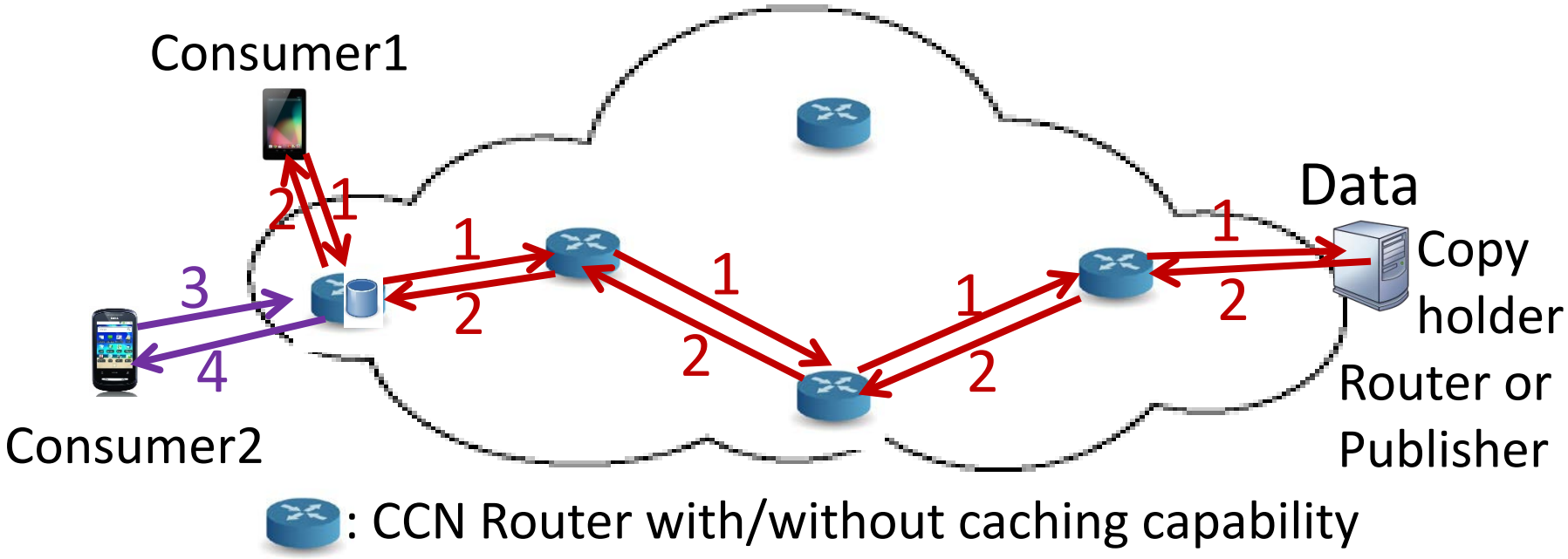
# In-Network Data Retrieval




From centralized cloud/server-based data provision
to distributed in-network caching-based data provision

# Content-Centric Network (CCN)

**Packet Types:** Interest/Data




**Publisher:** the entity that publishes data in network.

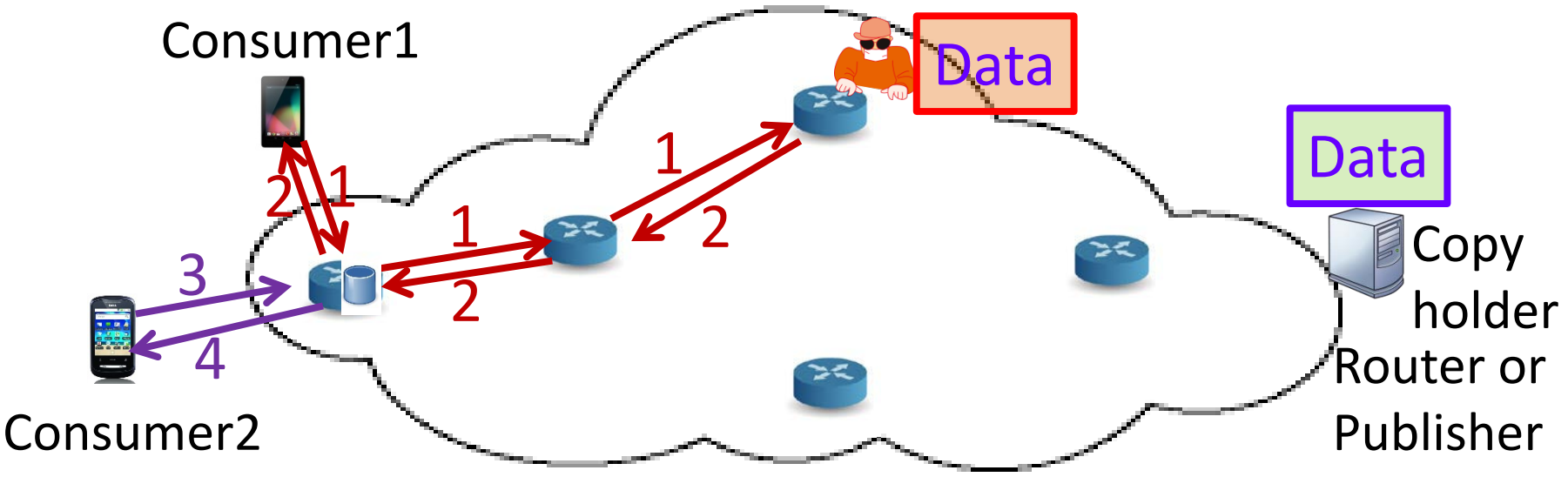
**Consumer:** the entity that retrieves data from network.

**Copyholder:** the entity that provides data to network. (Caching Router or Publisher)

# Adversary Model

- A1 (Content Poisoning Attack): Impersonate a copy holder to provide fake data
  - Currently, the content is only signed with the key of the entity who publishes it.
  - Impossible to verify the publishers' validity unless all routers use the authentication service of Certificate Authority (CA) for all forwarded/cached data
- A2 (Interest Flooding Attack): Impersonate a Consumer to request data
  - Much existing work on restricting the Interest sending rate
  - Impossible to verify the consumer's validity unless all the Copyholders (Router or Publisher) use the authentication service of CA

# Content Poisoning Attack




Fake/corrupted data are cached along the path

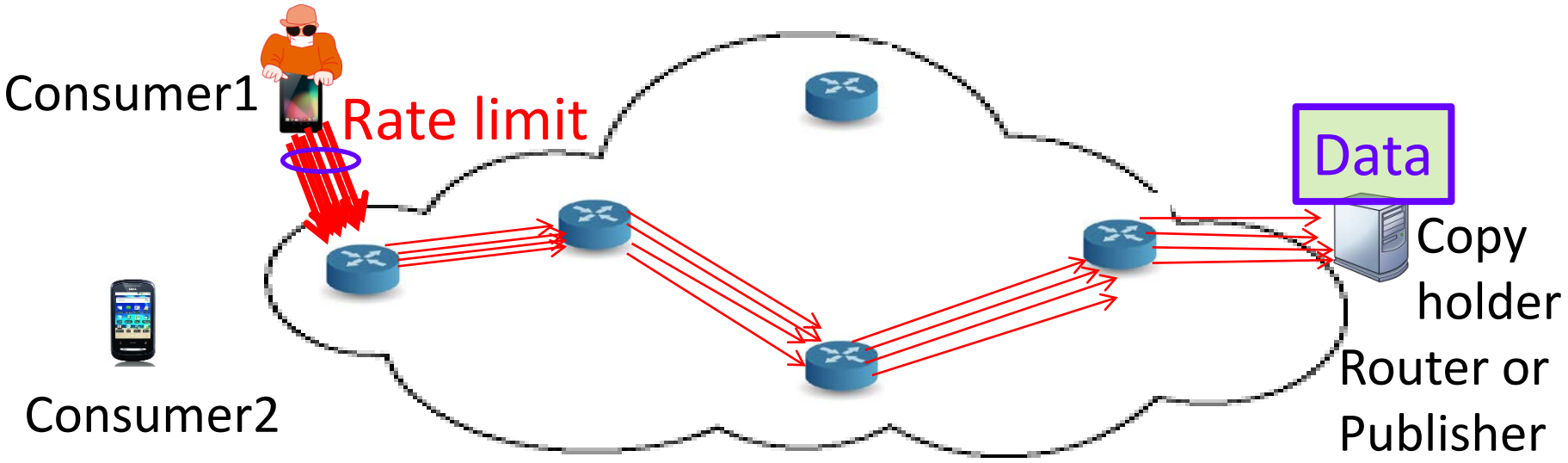
Consumers who use the path always retrieve the wrong data, because the router does not detect the cached data validity (as it's signed by attacker correctly)

Fake data are further cached, which pollute the routers as virus spreads.

Consumer2 and other potential consumers will also retrieve the fake data from routers.

Routers need to verify the data before caching. Consumers verify copyholder and path to identify the polluted entities besides data verifications.

# Interest Flooding Attack



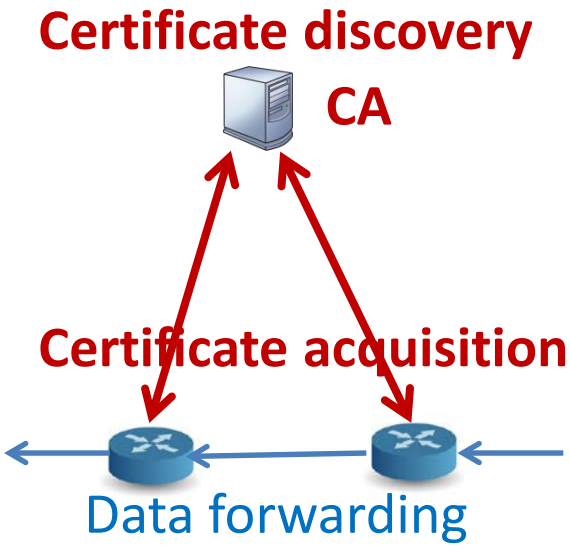

Consumer1 floods the Interests to the network to malfunction routers.

Even malicious Interests can be reduced by restricting rate, some malicious Interests still can reach the copyholder.

It will enhance the effect to inhibit the Interest flooding attack, if routers verify the Interests before replying the data.

# Traditional Authentication Approach

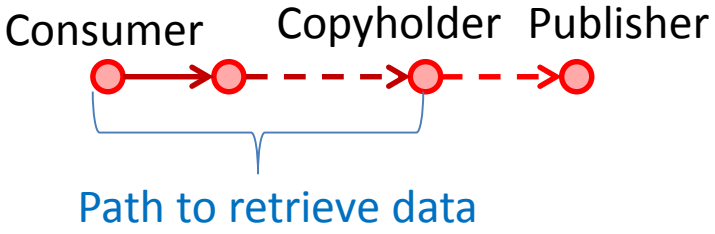
- CA-based PKI (Certificate Authority-based Public Key Infrastructure)
  - Routers rely on centralized server to discover the required certificates.
  - Routers need interact with CA to discover and acquire the corresponding certificates using additional messaging for authentication, which leads it to be infeasible during data retrieval.




- Scalability Problem
  - For router, it is highly cost that each router needs to verify the signature of every data and every Interest.
  - For consumer, the public key of all potential copy holders should be provided to users beforehand.

# Hop-by-Hop Authentication Mechanism (HopAuth)

- **Key idea**
  - HopAuth constructs a certificate chain between copyholder and consumer along the path
  - HopAuth can be CA-independent, which also can partially use CA-based trust to shorten the length of certificate chain and form a suspension-chain based on web-of-trust concept
  - Packet forwarding with collection of certificate chain without extra messages
  - Extend the entity trust graph with the application trust graph

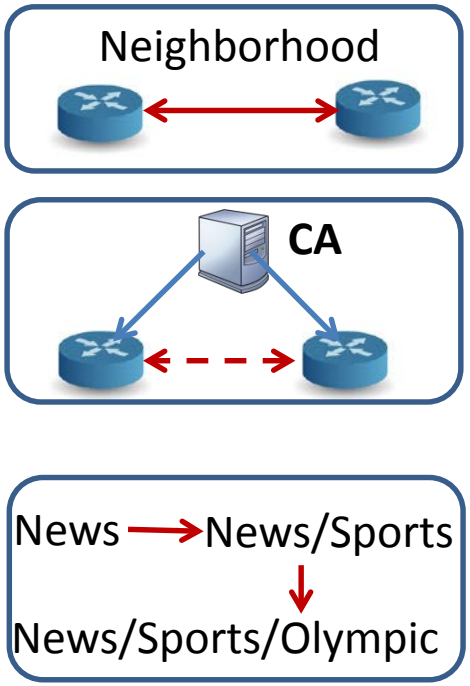



- **Effects**
  - Routers cache data only after authentication. Authentication can be performed offline. (Currently caching data without authentication)
  - Publisher authentication, copyholder authentication, consumer authentication, and path authentication can be achieved as required.

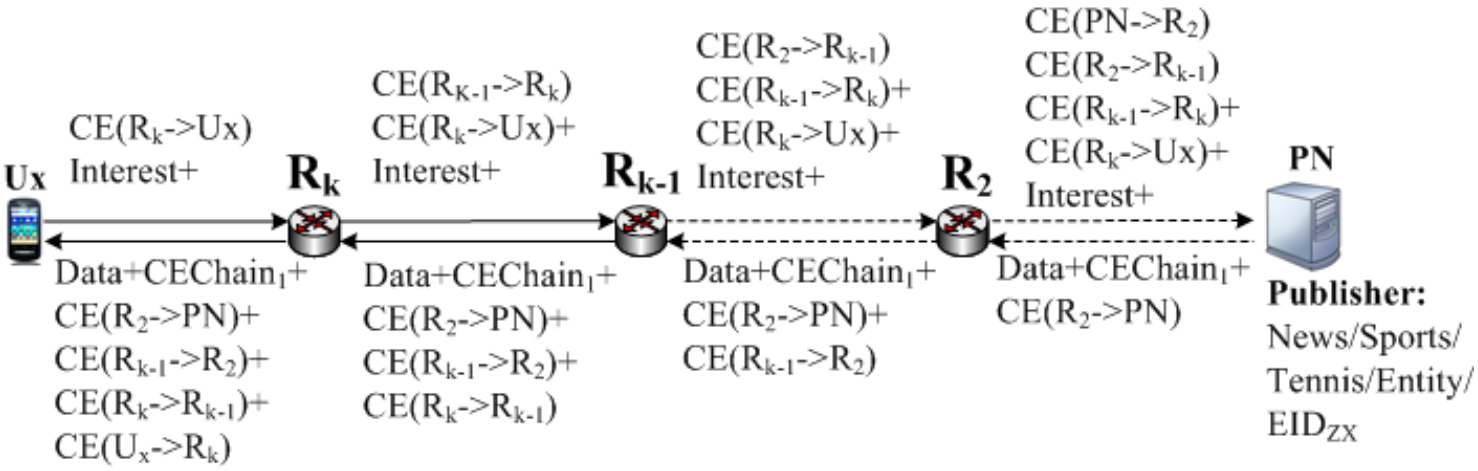
# Certificate Management

- **Physical entity:** the entity that communicates using a physical device. **Logical entity:** the entity that is involved in an application. (authorizer, sub-authorizer, publisher)
- In HopAuth, certificates are issued based on trust relationships, namely as neighbor-based trust, CA-based trust for physical entities, and authorization relationships in applications for logical entities.

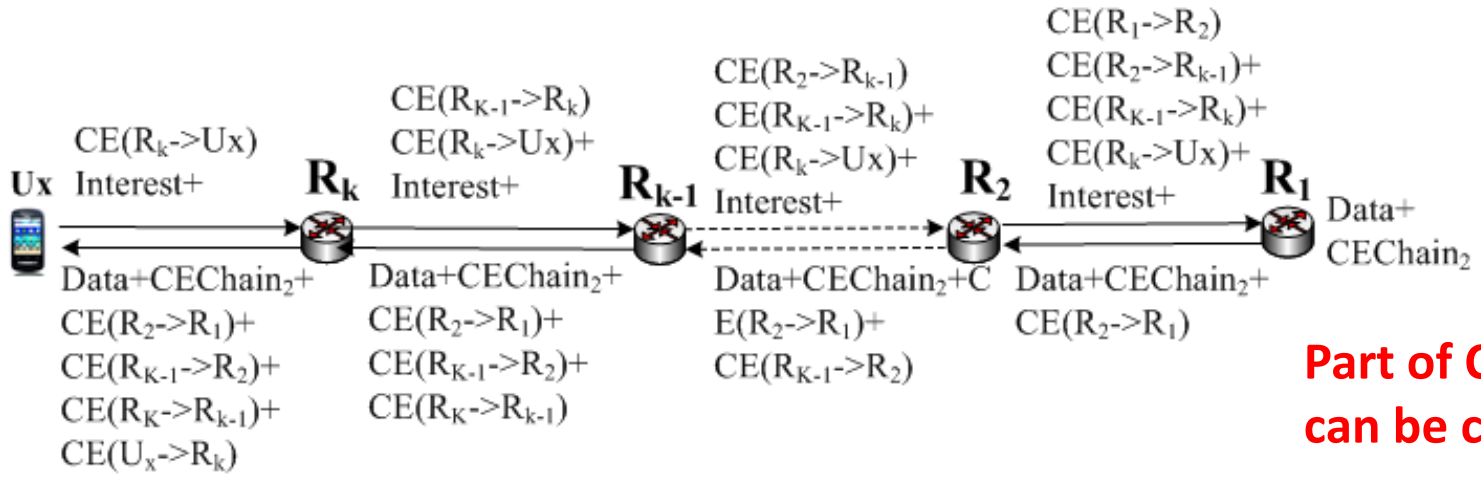
- Physical Entity Trust
  - Neighbor-based trust: Routers issue certificates to each other in the neighborhood.
  - CA-based trusts: CA issues certificates to highly trustable physical entities, which form highly trustable router group (HTRG).
- Logical Entity Trust
  - Authorization relation-based trust: If logical entity A authorizes the right for entity B to manage a sub-category or publish data, A should provide a certificate for the true public key of B.



# Forwarding-Integrated Authenticable Data Retrieval




**(a) Data Retrieval from the Publisher**



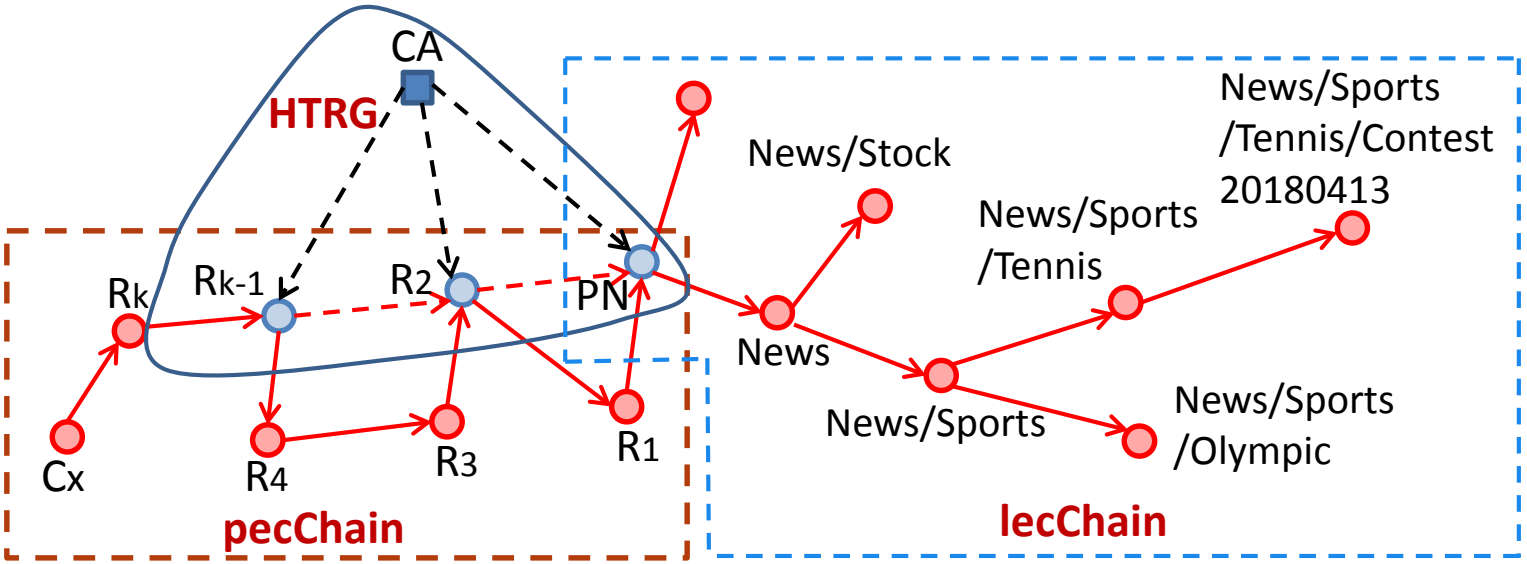

**Part of CEChain can be cached.**

**(b) Data Retrieval from the intermediate physical entity**

$CEChain_1$: Certificate chain from PN to Publisher

$CEChain_2$: Certificate chain from $R_1$ to PN and PN to Publisher

# Suspension Chain Model (SCM)




Cx: Consumer x  Ri: Router i  PN: Publisher Node  CA: Certificate Authority

: Neighbor-based trust relation  : CA-based trust relation in HTRG

:Suspension CA trust relation

**CA:** Certificate Authority  **PN:** Publisher Node

**HTRG:** Highly Trustable Router Group

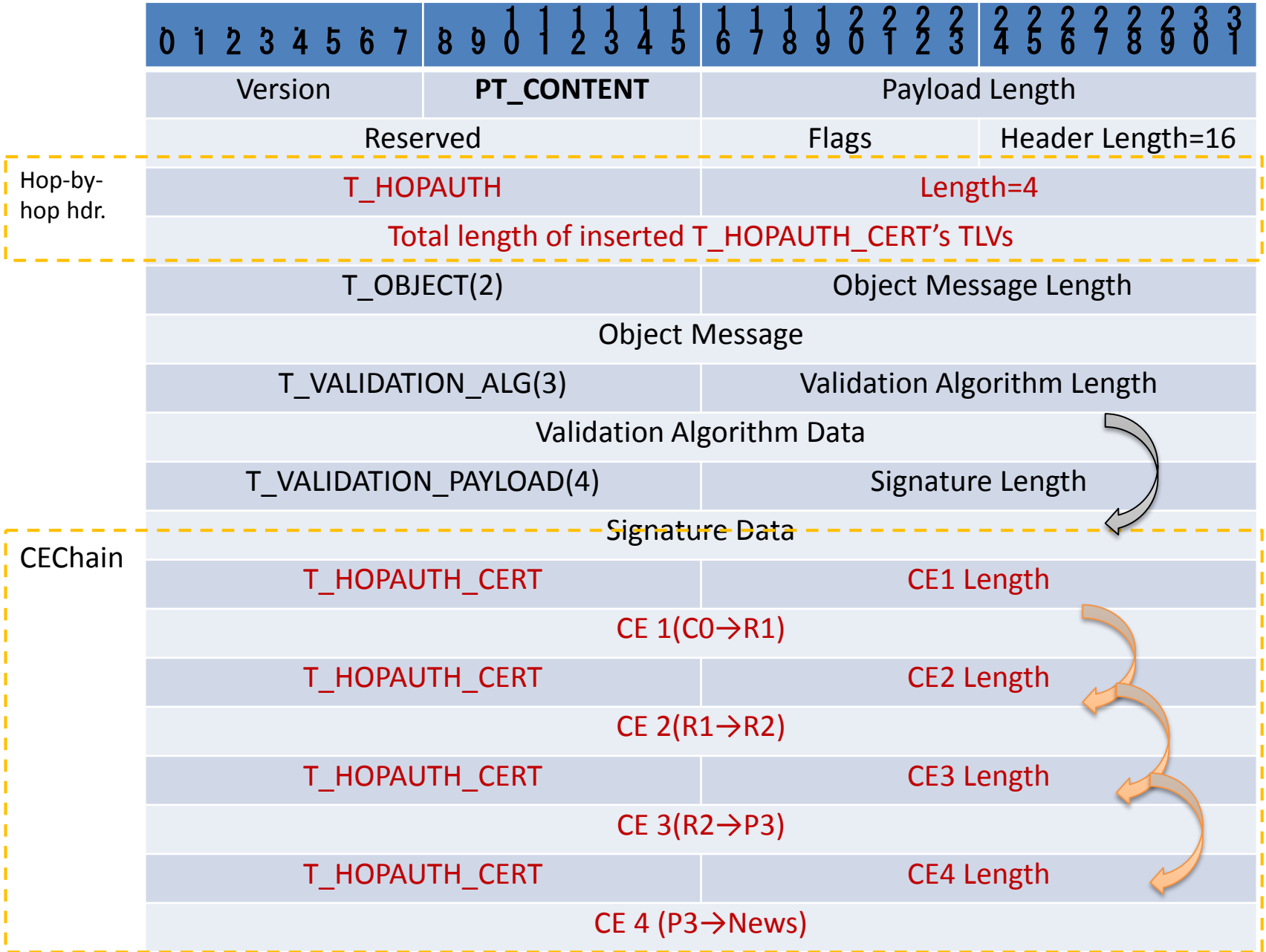
**pecChain**: Pyshical Entity Certificate Chain

**lecChain**: Logical Entity Certificate Chain

# Security Levels

- We identify **three security levels** as follows.
  - **Security Level 1 (Low):** Publisher Authentication
    - Authenticate the entity who publishes data.
    - For one piece of data, the suspension chain will be constructed only once.
  - **Security Level 2 (Medium):** Copyholder Authentication
    - Publisher Authentication + Authenticate the router who provides data.
    - For each copyholder during data retrieval, the suspension chain will be constructed once.
  - **Security Level 3 (High):** Path Authentication
    - Copyholder Authentication + Authenticate the path from which data are retrieved.
    - For the first time data chunk retrieval from one path, the suspension chain will be constructed.
    - For the following data chunk retrieval from the same path, a chain of certificate name will be appended.

# (Potential) HopAuth Packet Format



# Certificate Size

- DTLS (Datagram Transport Layer Security in Constrained Environments) [REF.1]
  - ECDSA P-256: **91 bytes**
  - ECDSA P-384: **120 bytes**
  - ECDSA P-521: **156 bytes**

For 2K bytes size data chunk, 546 bytes (26.7%) are used to accommodate 6 certificates, if ECDSA P-256 is employed.

| RSA Public Key Length (bit) | ECDSA Public Key Length (bit) |
|---|---|
| 1024 | 160 |
| 2048 | 224 |
| **3072** | **256** |
| 7680 | 384 |
| 15360 | 512 |

[REF.1] Datagram Transport Layer Security in Constrained Environments
https://www.ietf.org/proceedings/83/slides/slides-83-lwig-2.pdf

ECDSA: Elliptic Curve Digital Signature Algorithm

# Conclusions

- As an authentication protocol, we propose **HopAuth** to enable hop-by-hop certificate collection and authentication.
- As a trust model, we propose **SCM**, which merges web-of-trust and partially CA-based Trust to construct a suspension chain to enable content-centric trust.

Thank you!