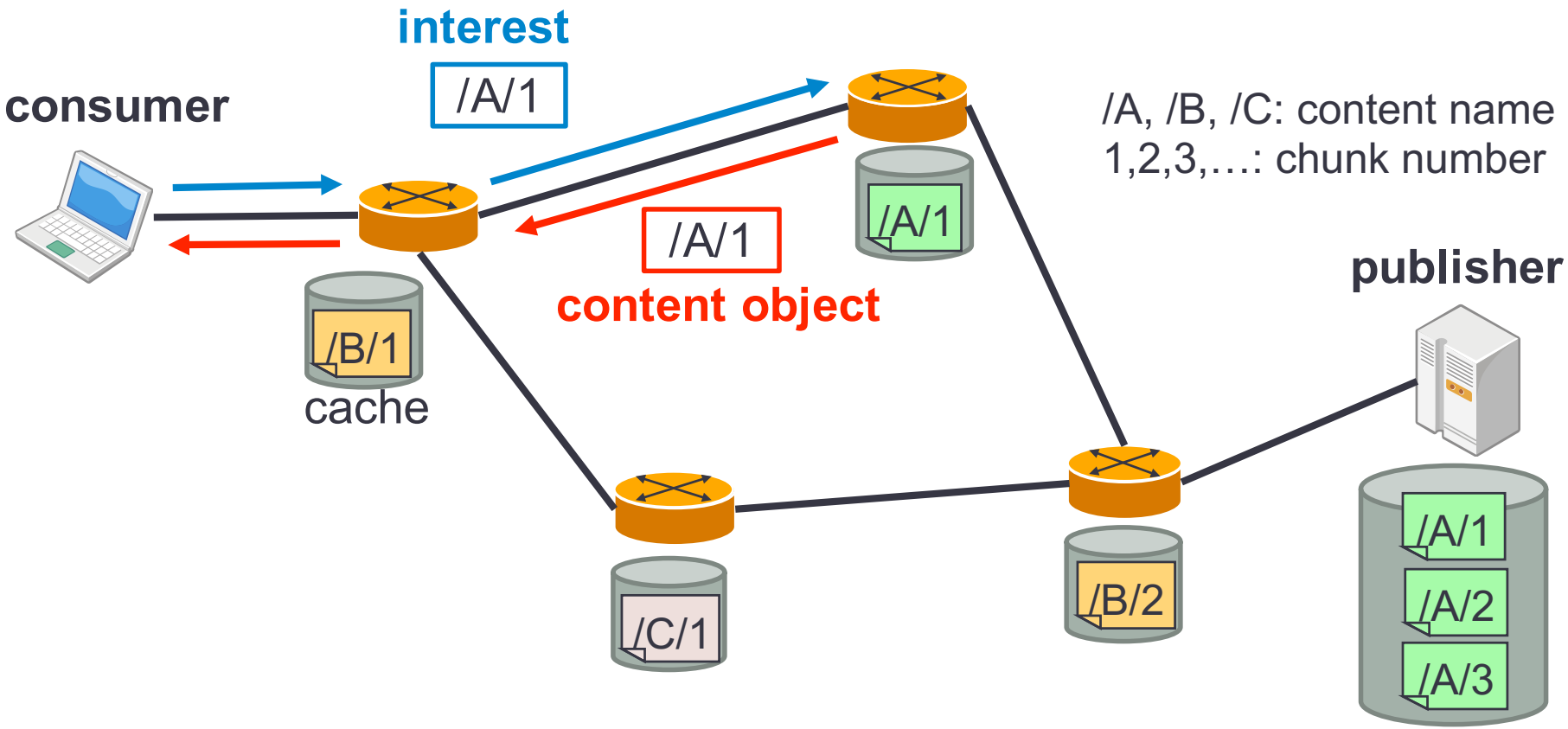
An Encryption-Based Access Control Framework for Content-Centric Networking

Jun Kurihara[1], **Ersin Uzun**[2] **and Christopher A. Wood**[2,3]

[1]KDDI R&D Labs., [2]PARC, [3]UC Irvine

IFIP Networking 2015
Toulouse, France, May 20, 2015

Content-centric networking (CCN)



- Content-oriented naming and routing
- Exchange of interests (request) and named content objects (response)
- In-network caching of content objects

CCN 1.0 = The latest version of CCN architecture.

Content protection in networking architectures

Existing host-to-host Internet design:

Network messages are coupled with their source/destination nodes

⇒ Messages always arrives from their original server.

⇒ Secure point-to-point channels[1] have been considered. (Session-based access control)

CCN:

Network messages are independent from source/destination

⇒ Messages may NOT arrive from the original publisher due to in-network caching.

⇒ Content must be encrypted so as to prevent invalid disclosure by unauthorized users. (Encryption-based access control)

[1] e.g., TLS

In this talk, we present a novel encryption-based access control framework for CCN 1.0 (**CCN-AC**).

Agenda

1. Elements in CCN 1.0
2. Overview of CCN-AC
3. Example of instances
4. Conclusion

Network messages in CCN 1.0

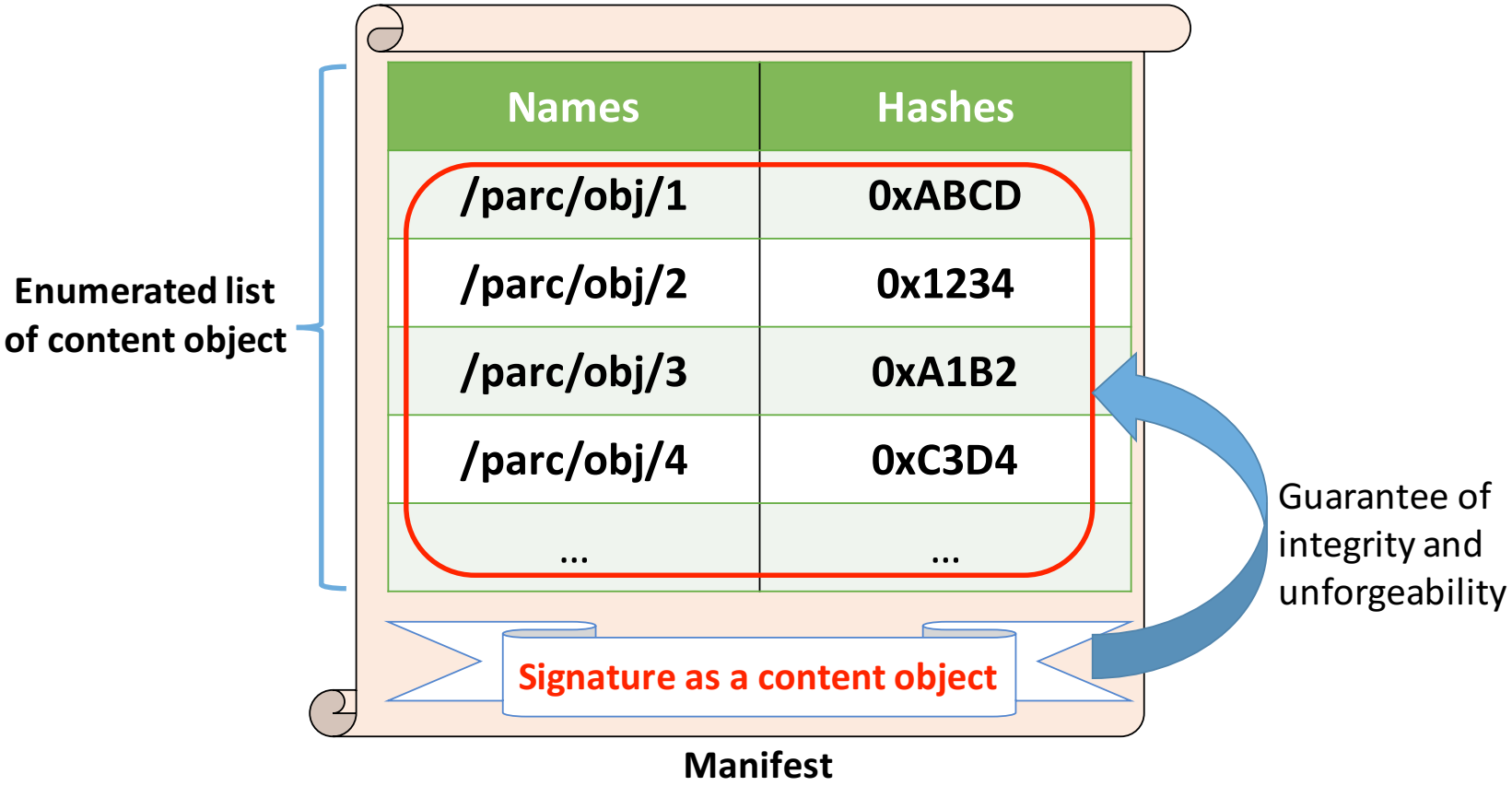
Messages = Interests and content objects

- Every message can optionally include the signature for the authentication of the payload.
- Interest can optionally specify
  - the hash of the content object requested by the interest
  - the signature key ID of the content object publisher

  in addition to the name.

Manifest in CCN 1.0

A manifest is a type of content objects that has a signature



- A manifest provides an enumerated list of names and hashes
- The signature guarantees the integrity and unforgeability of listed items

Content retrieval through manifests

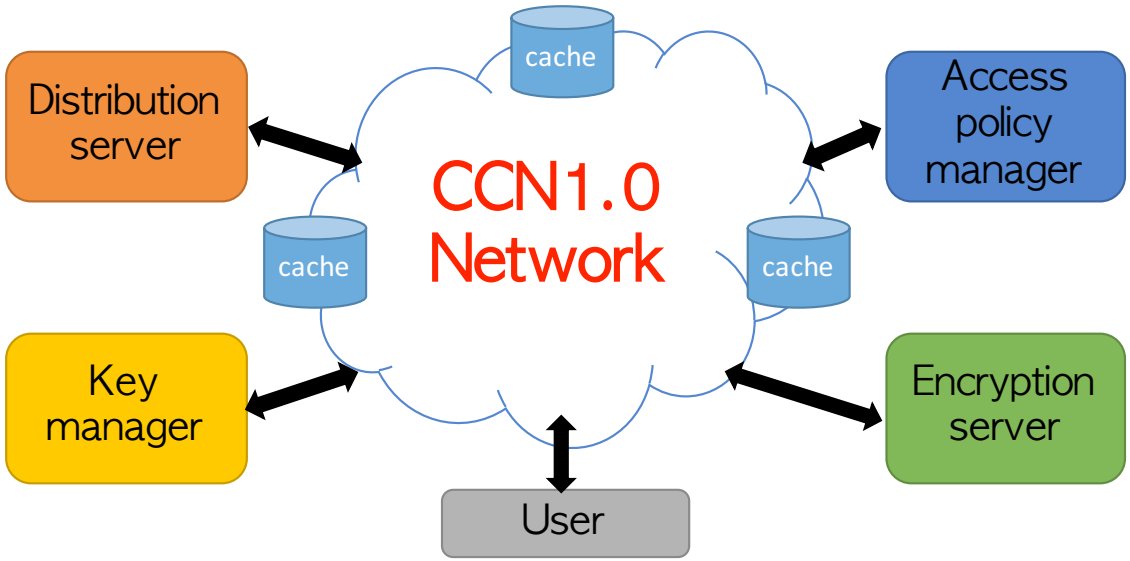
1. A consumer retrieves a manifest
2. He issues interests by the information in the manifest
3. He retrieves content objects, and checks their validity by hashes in the manifest

Observe:

Hashes of content objects are protected in the manifest
$\Rightarrow$ No signature is needed in content objects listed by a manifest for their integrity

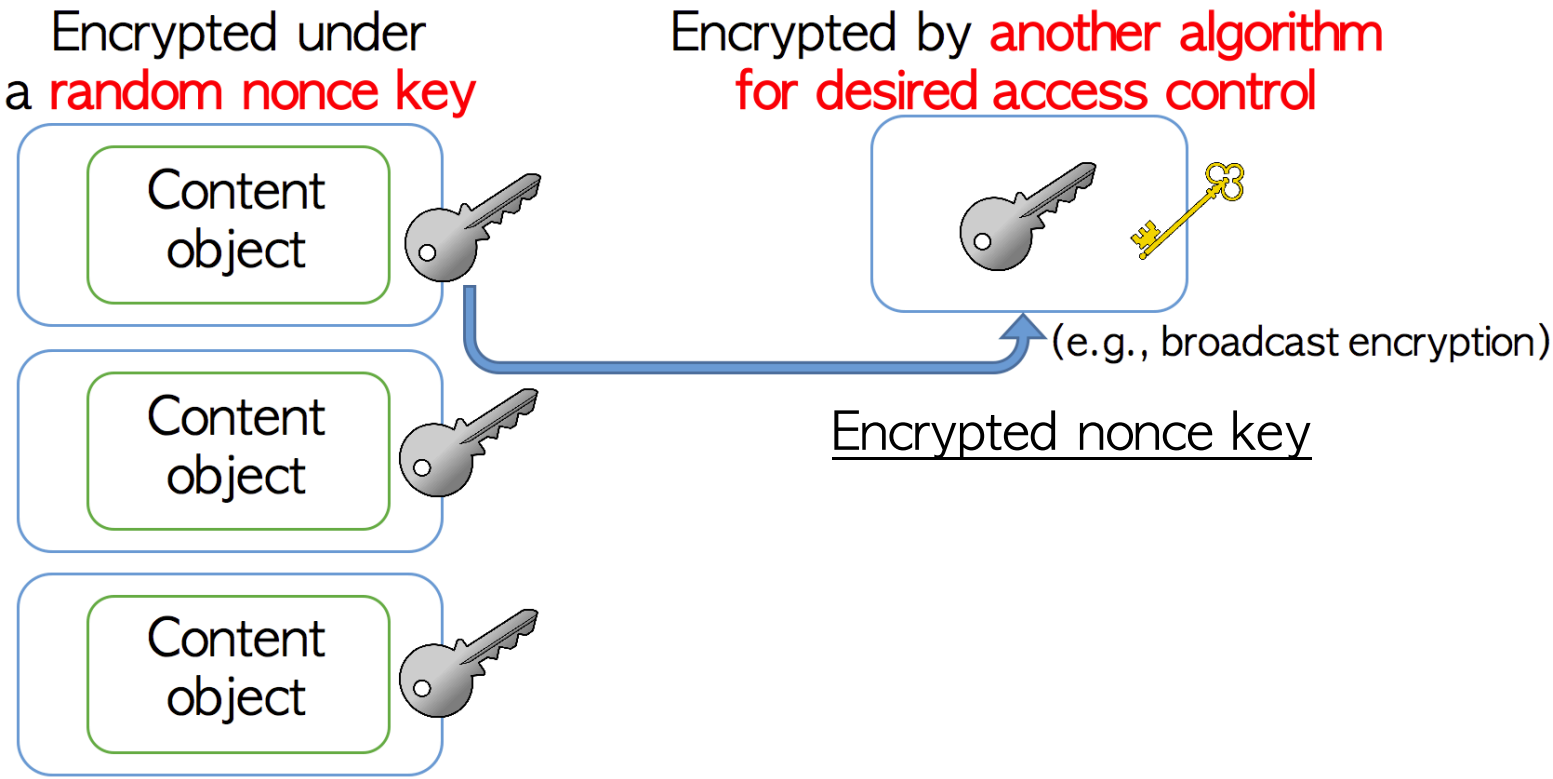
Design principles of CCN-AC

- Provide a comprehensive and scalable framework for encryption-based access control in CCN.
  - Any access control policy and scheme can be instantiated
  - Any number of producers and consumers can be supported
- Maximize the usage of in-network caches for the access controlled contents



Envisioned System Design

CCN-AC approach (1/4): Hybrid encryption of content data

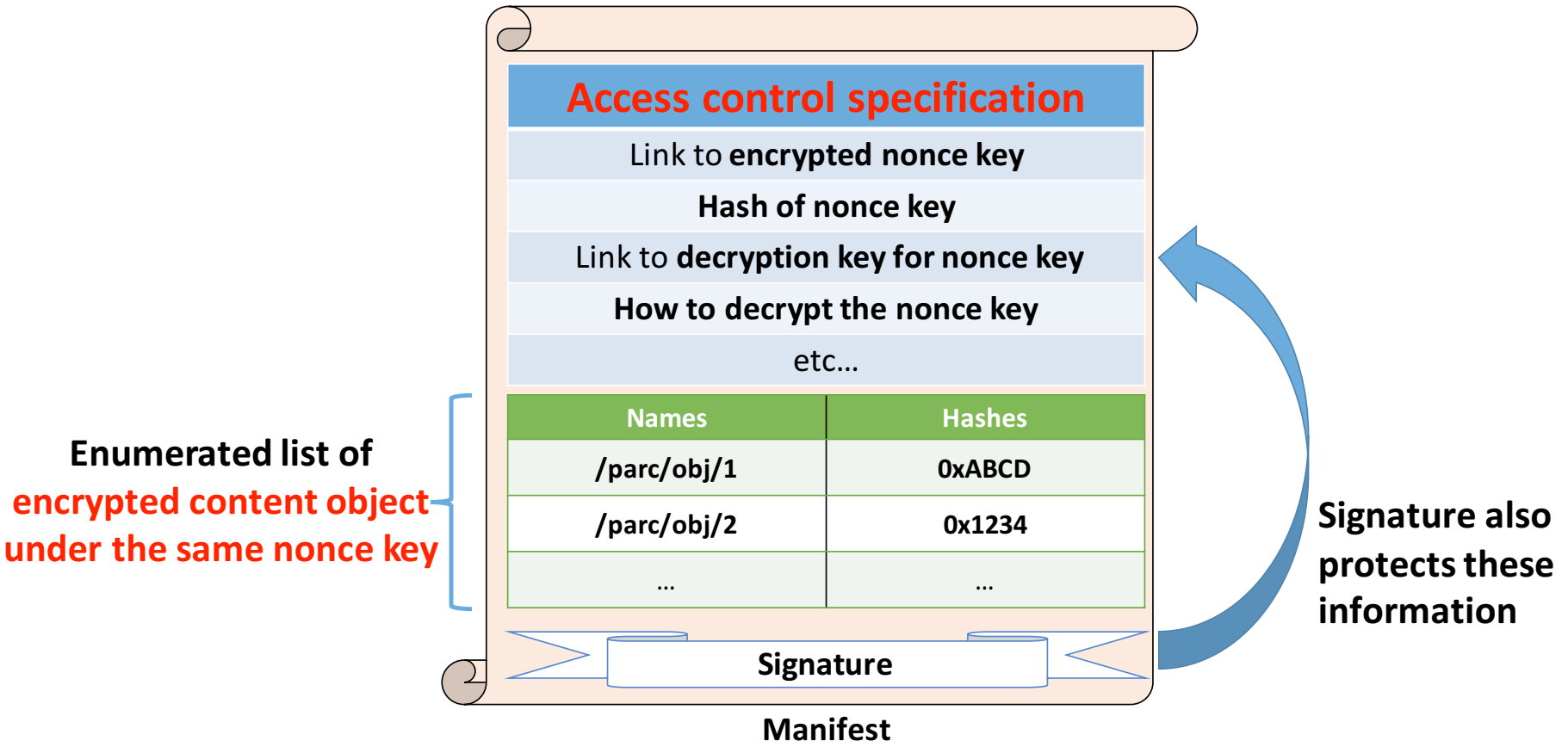


-> Can be recycled for every user

-> Still can use in-network cache

CCN-AC approach (2/4): Manifest-based enforcement

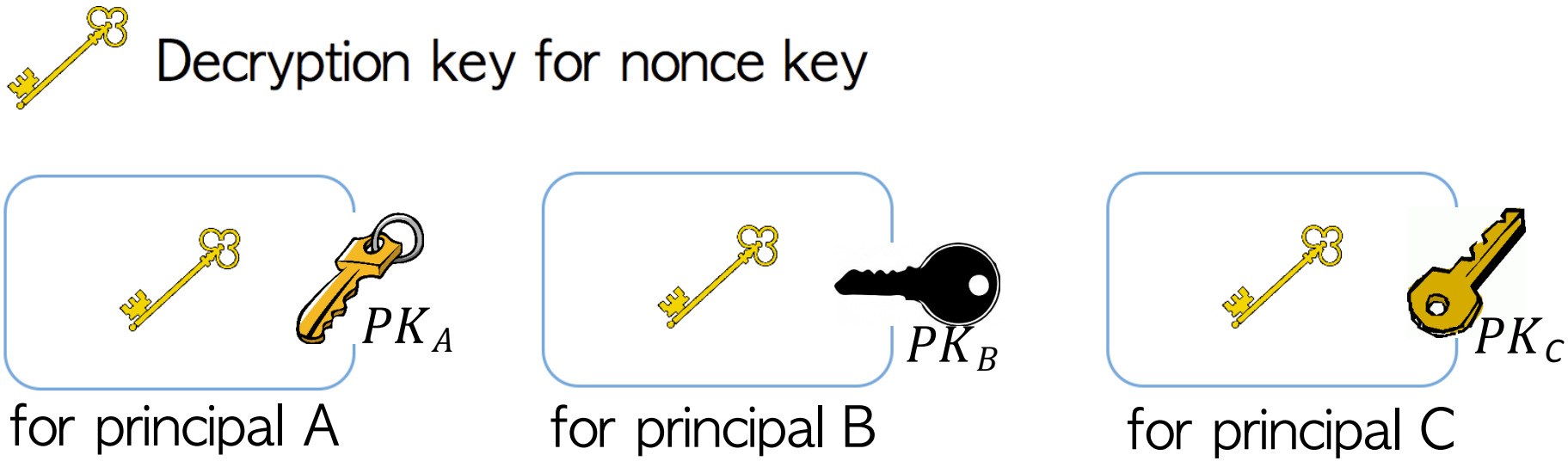
- Content objects listed in a manifest are encrypted under the same nonce key
- All information needed to decrypt listed content objects can be obtained through the manifest



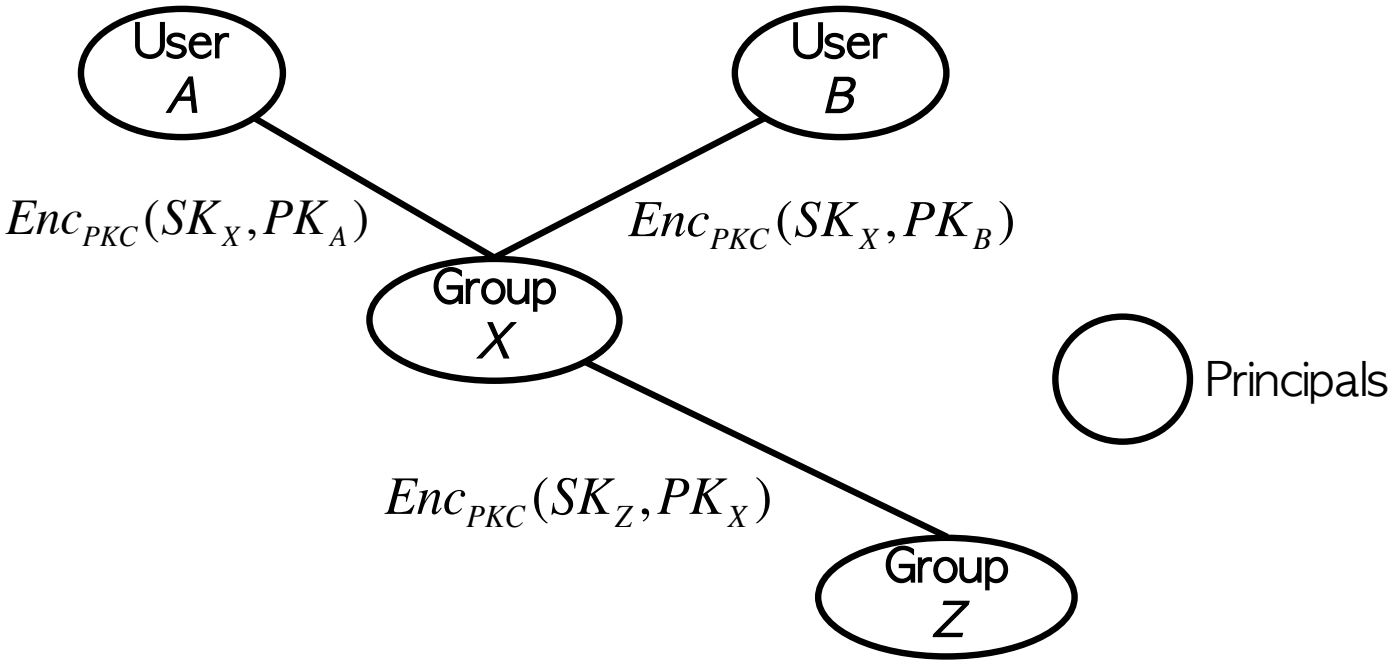
CCN-AC approach (3/4): “Principal”-based access control

Principal = Individual or group of users

A unique pair of public key PK_i and private key SK_i is assigned to each principal i.



All the encrypted keys are located in the network as separated content objects.



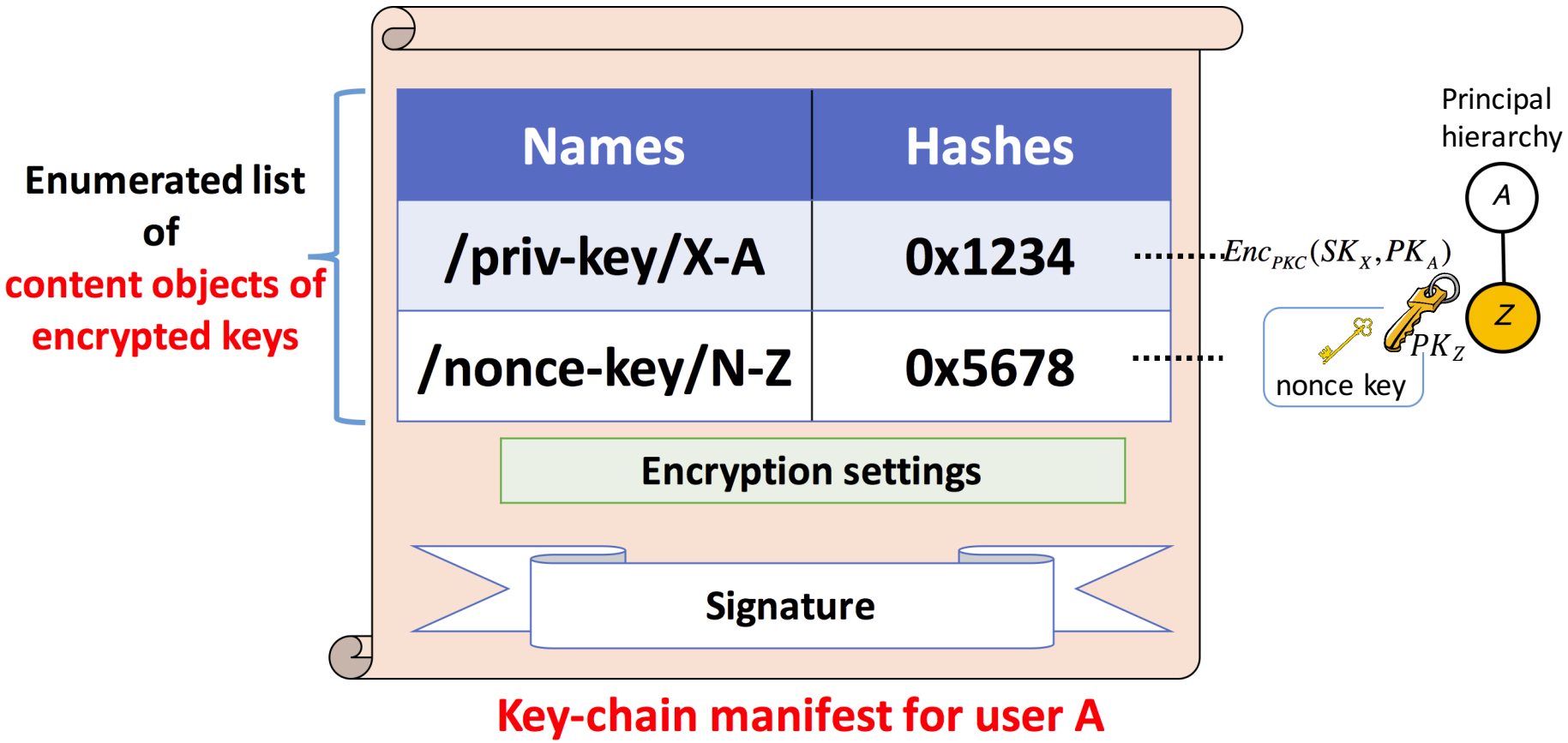
Each principal's private key is encrypted with its children's public keys

Only a qualified user can disclose the encrypted nonce key by traversing the principal tree.

CCN-AC approach (4/4): Key-chain

Each qualified user retrieves its required keys through a **key-chain**.

ex) Key-chain for user $A \in$ group Z.



A key-chain is an enumerated list of all encrypted keys for a user.

Summary of CCN-AC approach

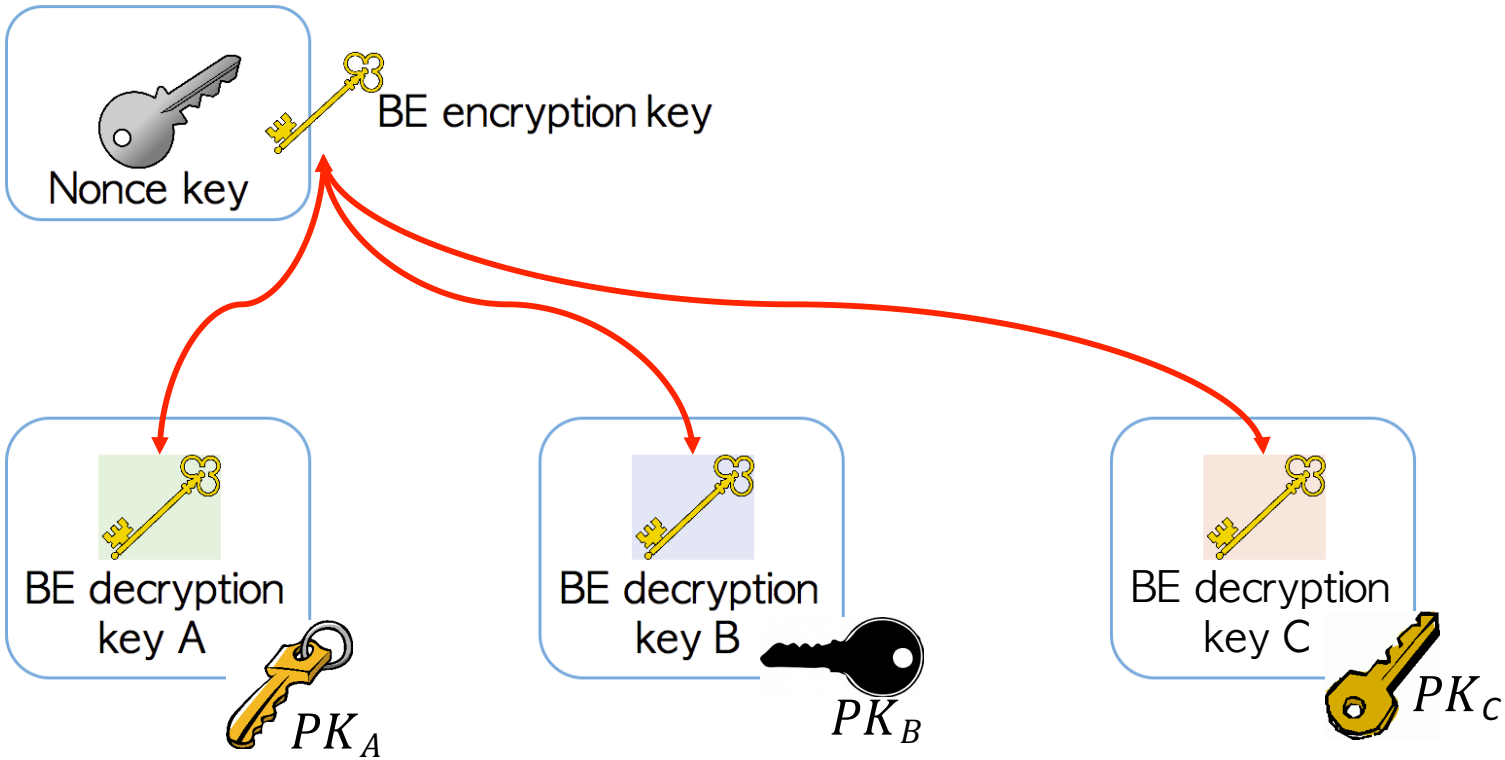
- Hybrid encryption of content objects.
- All information of access control can be obtained through manifests.
- Principal-based access control.
- Key-chain manifest provides all the required keys for each qualified user.

Access control instance using broadcast encryption

Assumption: Individual users are directly qualified to obtain contents.

Setting:

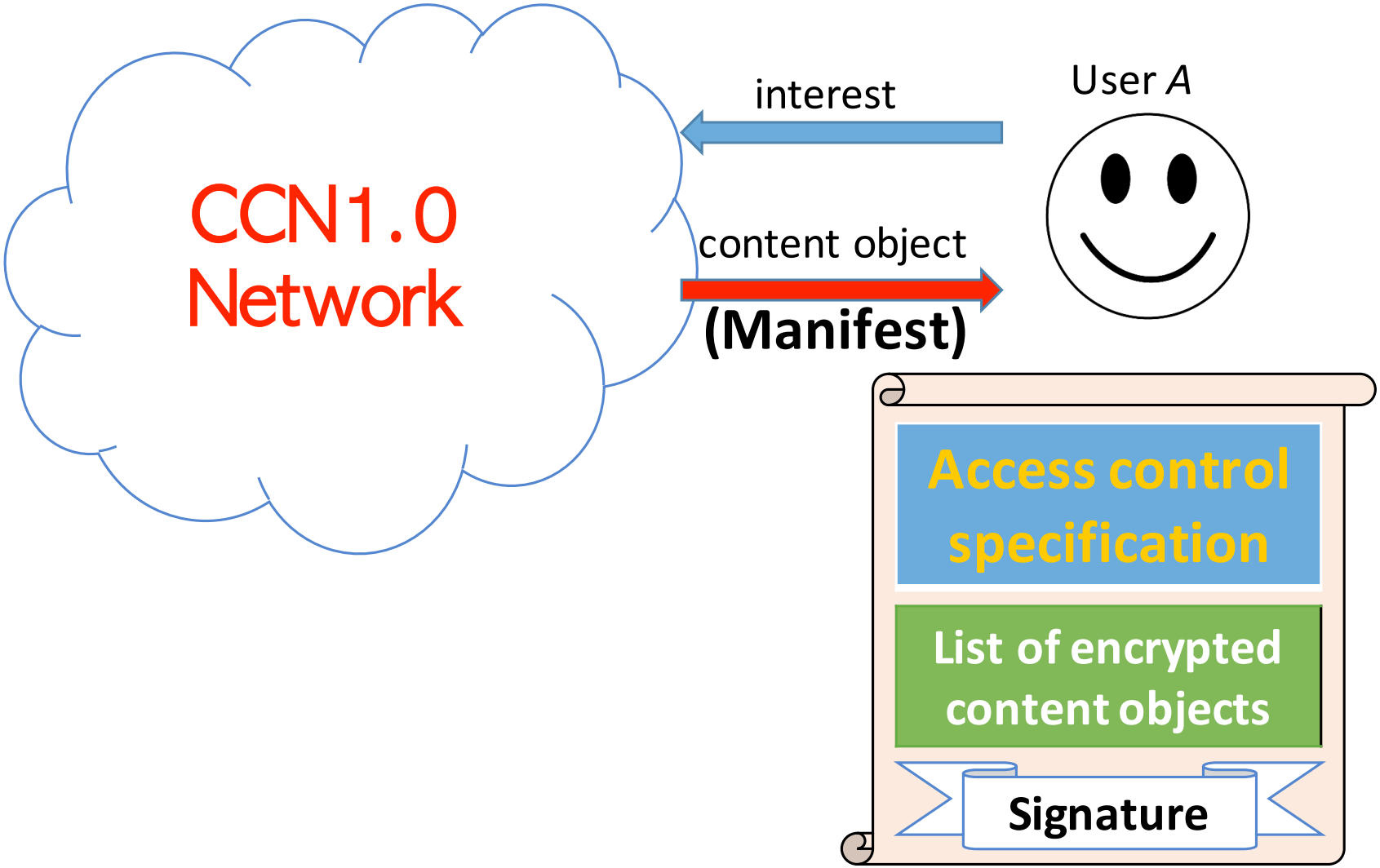
Nonce key is encrypted by a broadcast encryption (BE)



Each BE decryption key is encrypted under each user's public key

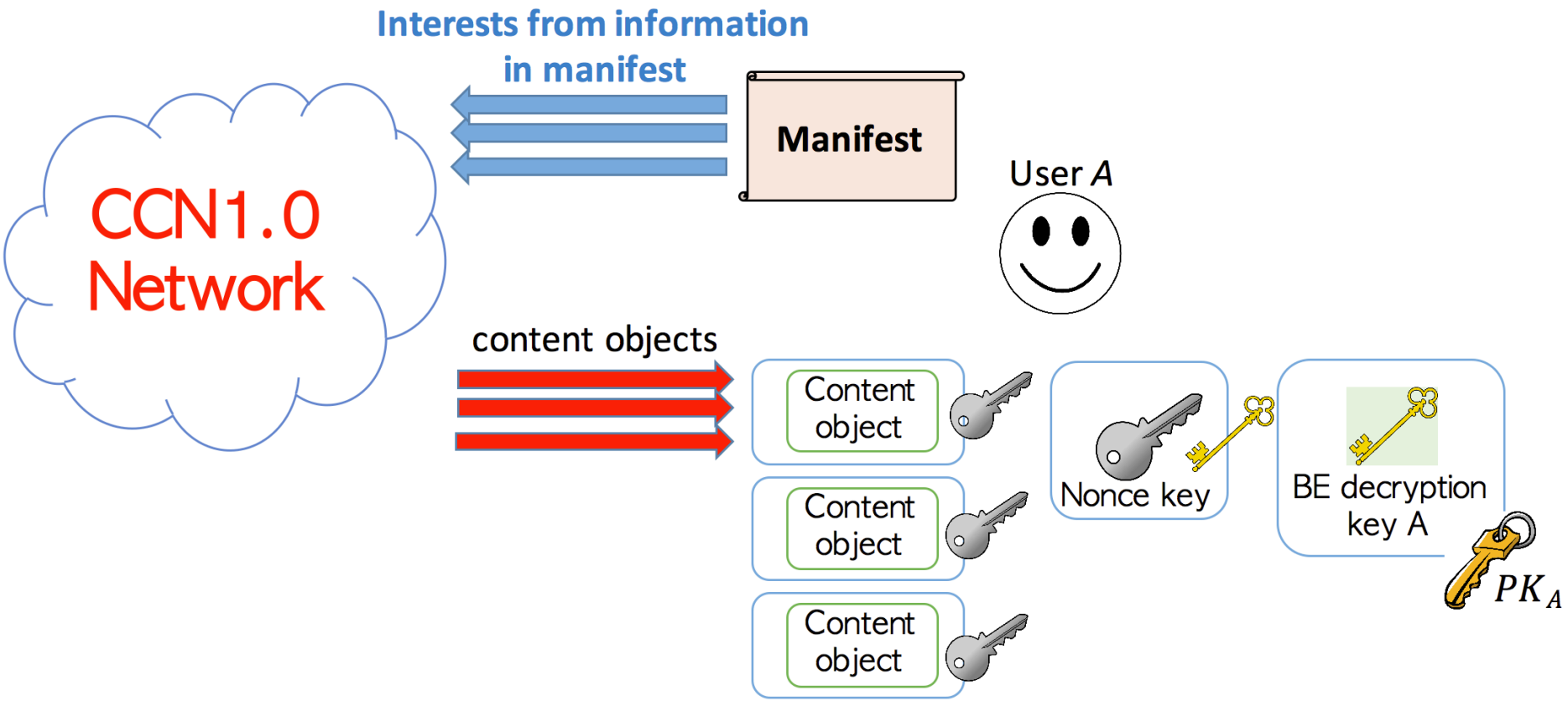
Retrieval of objects and decryption:

1) A user (User A) first retrieve the manifest.



2) Through the manifest, the user A retrieves

- All encrypted content objects
- BE-encrypted nonce key
- BE decryption key (encrypted under his public key PK_A)



3) By the retrieved objects, the user can obtain plaintext content objects.

If groups are qualified principals, the key-chain manifest is retrieved through the manifest and used to retrieve all required keys.

Other instances

- Group-based access control
- Attribute-based access control
- Proxy re-encryption-based access control
- Interactive protocols (TLS-like access control, etc)
- etc.

We will describe them in the ArXiV preprint of this paper.
(`http://arxiv.org/cs.NI/...`)

Conclusion

We proposed CCN-AC:

- A flexible, scalable and extensible encryption-based access control framework for CCN 1.0.
- Based heavily upon the CCN 1.0 manifest.
- Maintains the cache usage efficiency for sensitive content objects

Appendix

Key-chain is personalized

* Each user can retrieve his manifest through the access control specification of the root manifest.
* Each key-chain is personalized to each user, and has the personalized name (e.g., public key's hash) as well.