

Interest-Based Access Control in CCN

Cesar Ghali, Marc A. Schlosberg, Gene Tsudik, **Christopher A. Wood**

Department of Computer Science
University of California Irvine
woodc1@uci.edu



Agenda

1. Introduction and Access Control Overview
2. IBAC Security Model
3. IBAC via Name Obfuscation
4. Security Considerations
5. Experimental Assessment
6. Conclusions and Recommendations



CCN Elevator Pitch

- Content is named and transferred through the network from producers to consumers upon request
- *Any consumer* can ask for content by name
- Producers are responsible for access control



Notation

N name of a Content Object

$CO[N]$ content object with name N

$\mathbb{U}(N)$ set of consumers authorized to access content with name N

$\bar{\mathbb{U}}(N)$ complement of the above

\mathbb{G} group of consumers



The Access Control Problem

Question: How to ensure that only **authorized users** access a content object?

1. **Content-based:** Ensure that only authorized consumers can decrypt content they retrieve
2. **Interest-based:** Ensure that consumers can only retrieve content they are authorized to access



Content-Based Access Control

Main Idea: If $C_r \notin \mathbb{U}(N)$ then C_r should not be able to decrypt $CO [N]$

- A preliminary specification was first introduced in [1]
- Many variations based on different public-key cryptographic algorithms have been proposed (see [2]):
 - Broadcast encryption
 - Attribute-based encryption
 - Proxy-based re-encryption
 - ... etc.

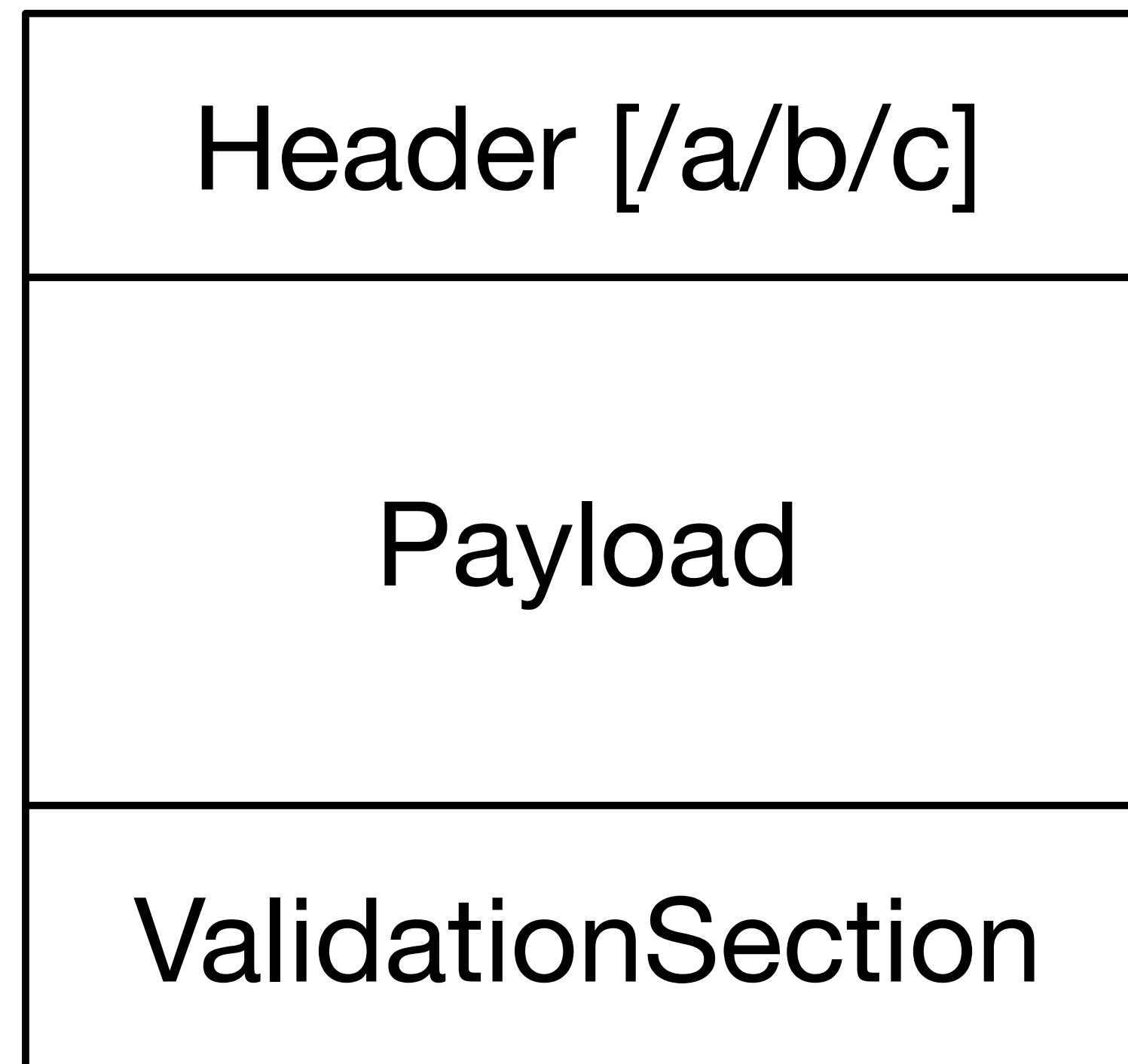
[1] Smetters, Diana, Philippe Golle, and Jim Thornton. CCNx Access Control Specifications. Technical report, PARC, 2010.

[2] Kurihara, Jun, C. Wood, and Ersin Uzun. "An Encryption-Based Access Control Framework for Content-Centric Networking." IFIP, 2015.

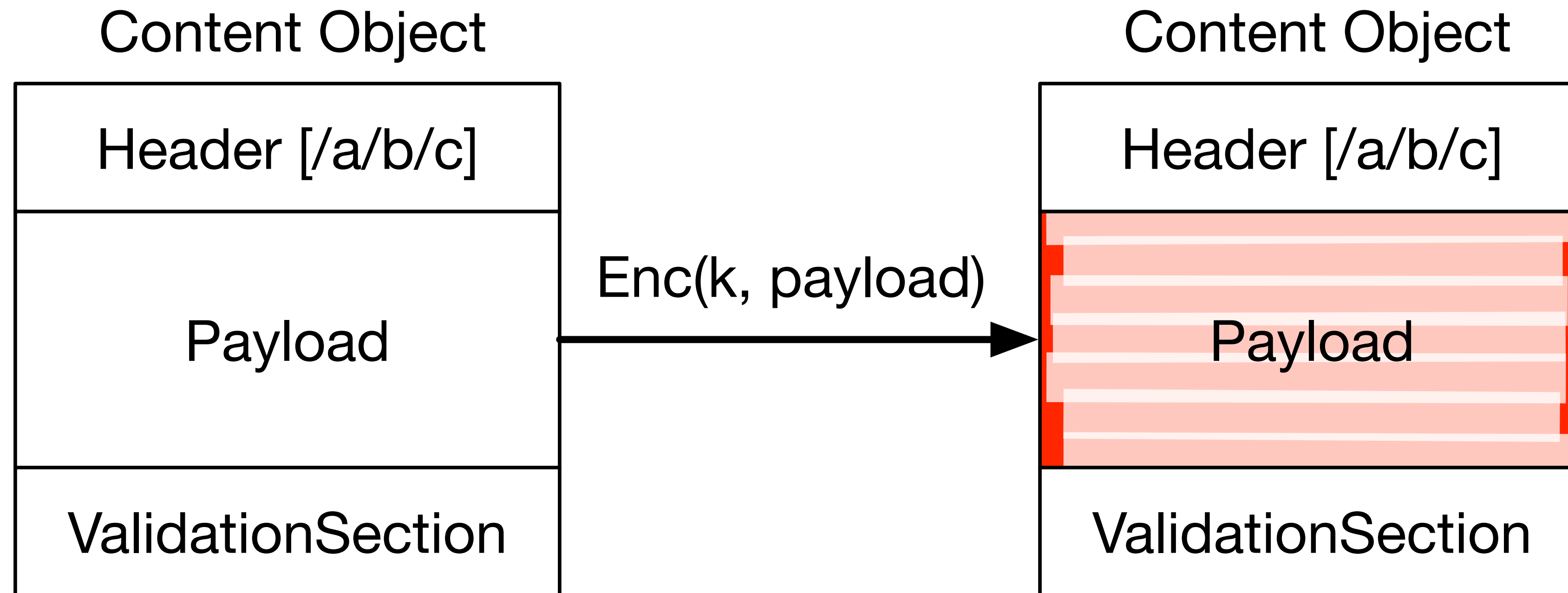


Content-Based AC in Pictures

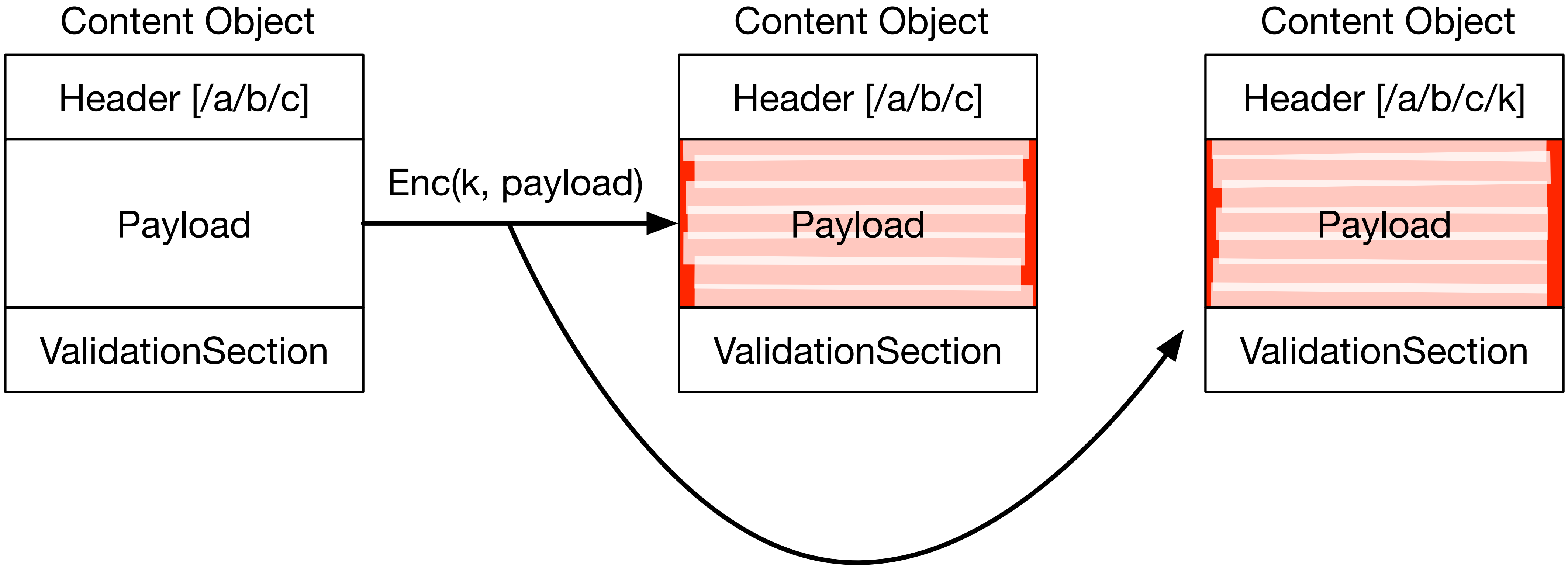
Content Object



Content-Based AC in Pictures (cont'd)



Content-Based AC in Pictures (cont'd)



Interest-Based Access Control

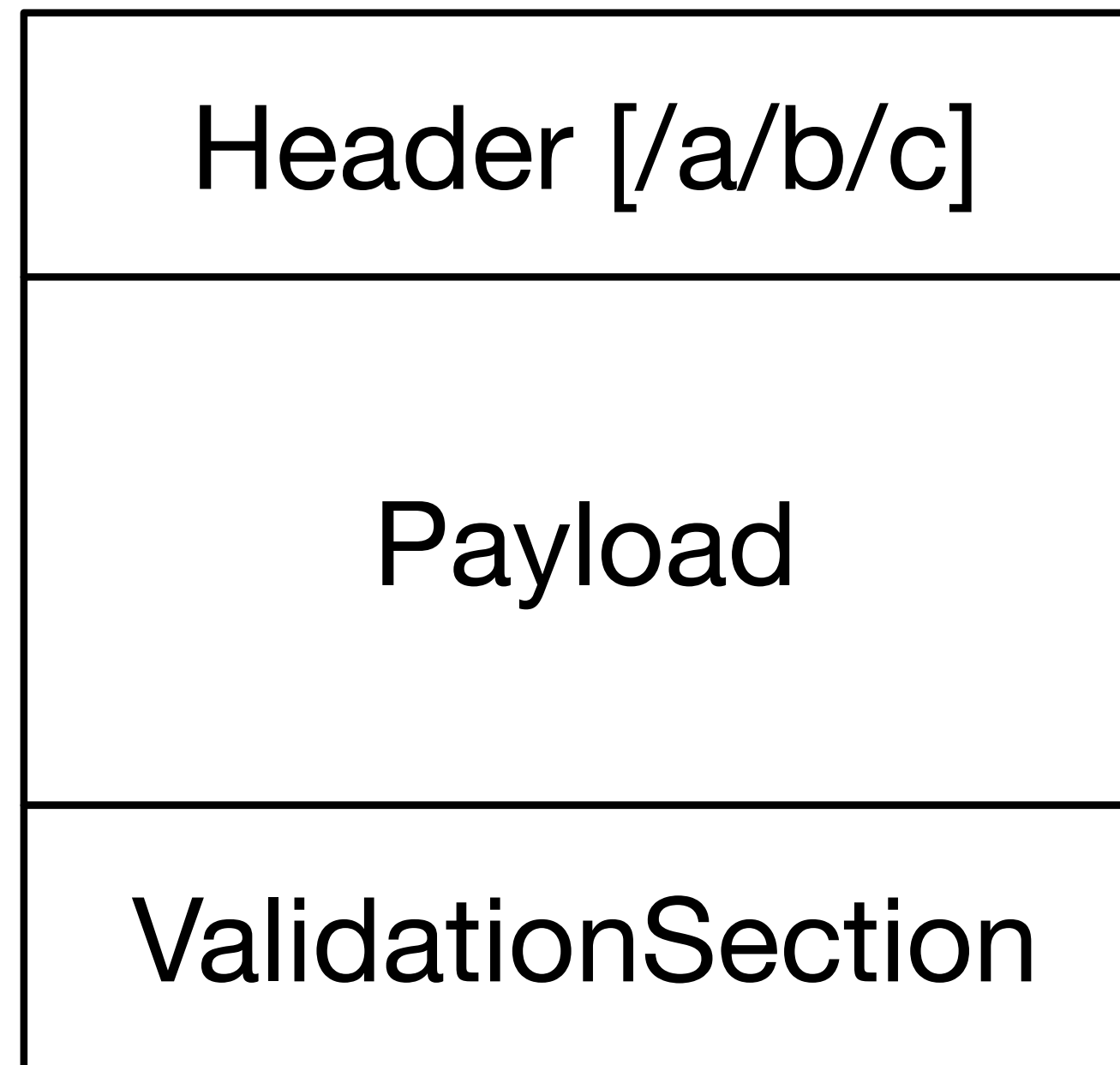
Main Idea: If $C_r \notin \mathbb{U}(N)$ then C_r should not be able to construct a correct interest for $CO[N]$

Implication: Interest names should depend on a secret that *only authorized consumers know*

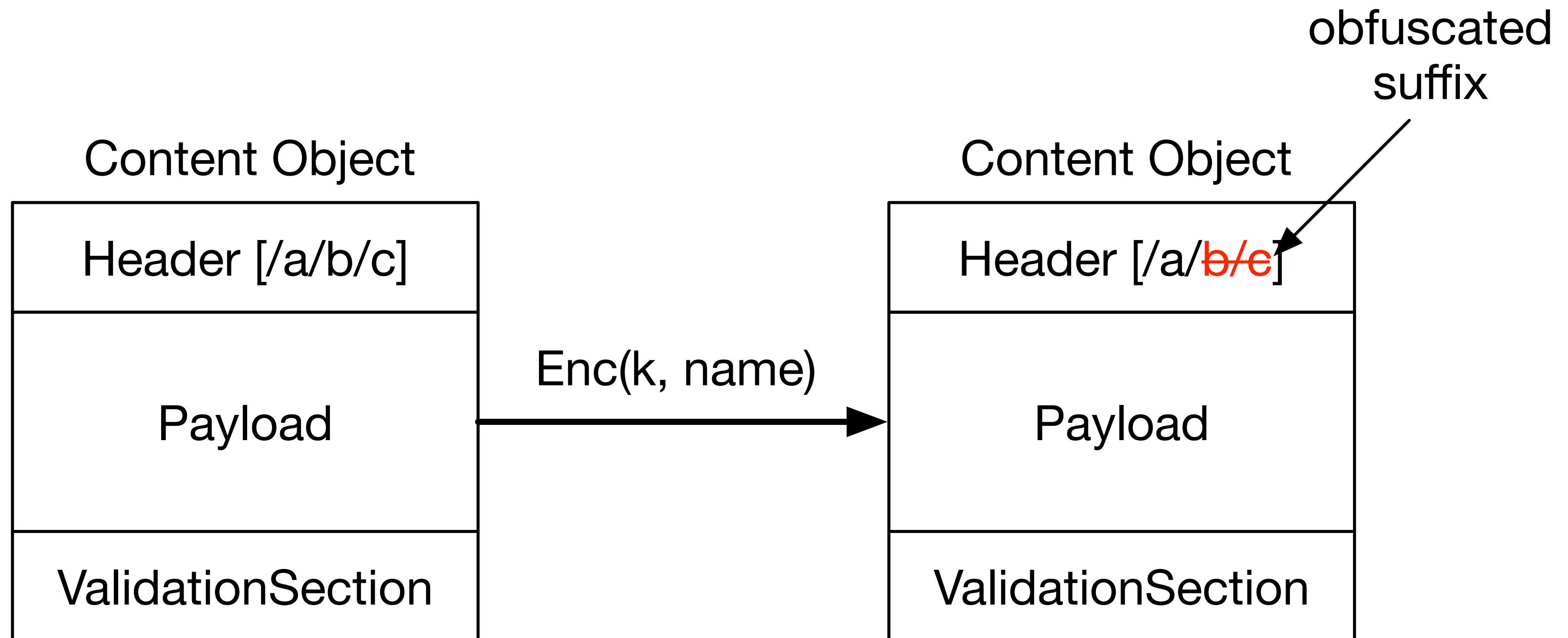


Interest-Based AC in Pictures

Content Object



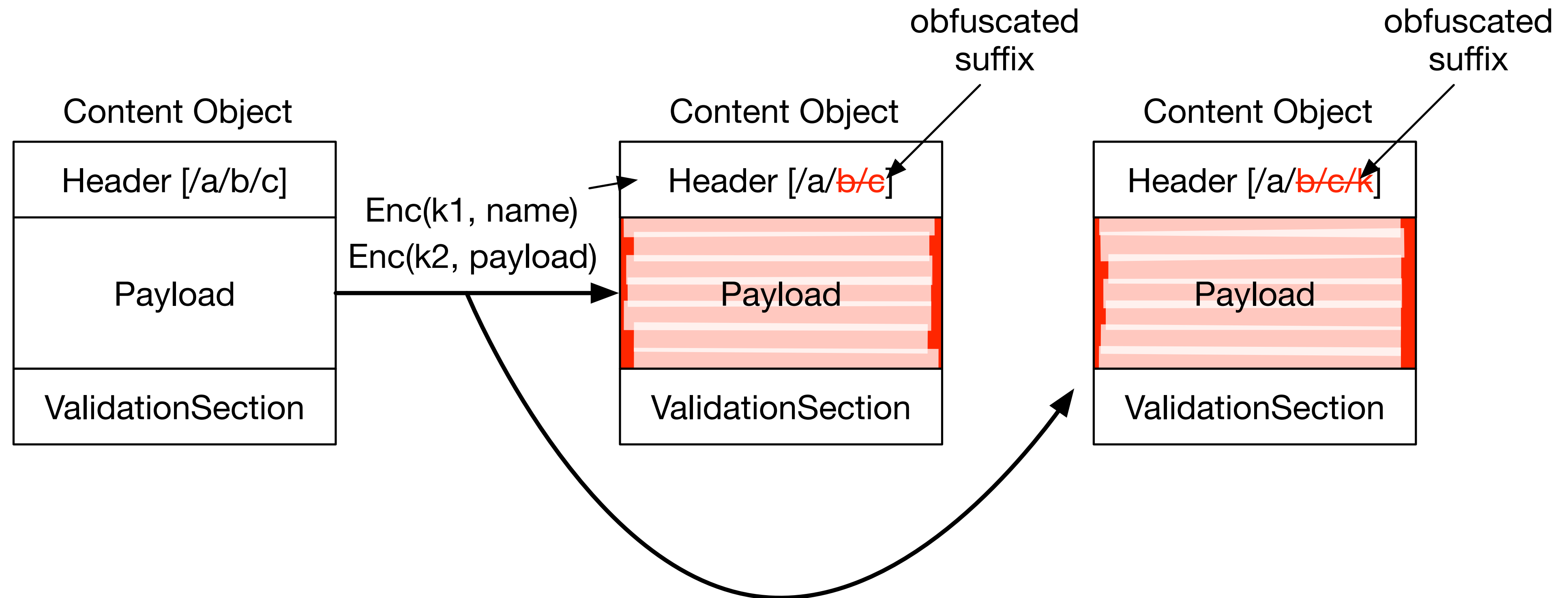
Interest-Based AC in Pictures (cont'd)



... why not do both?



Two Dimensions of AC



Security Model

IBAC is about **obfuscating the name** (the payload may also be encrypted...)

Security means: an adversary without the (group) secret cannot generate the same obfuscated name

Let $\text{Path}(C_r, P)$ be the set of all routers on the path between C_r and P

Assume **Adv** who can deploy and compromise any unauthorized consumer or any router $R \notin \text{Path}(C_r, P)$

- On-path adversaries can see the names in interest and content
- ...will consider this later



IBAC via Name Obfuscation

The goal of IBAC is to make the name N of a content object available under the name $N' = f(N)$ for some obfuscation function f

At least two ways to do this:

- Encryption-based
- Hash-based

Note: the obfuscation function only masks the suffix of a name — not the routable prefix



Encryption-based Obfuscation

$$N' = \text{Enc}(k, N)$$

where k is the private key associated with an authorized Cr



Supporting Multiple Groups

Question 1: What if we want group-based access control, i.e., where consumers in the same group generate the same obfuscated name?

(One) Answer: Consumers in group \mathbb{G} share the encryption key $k_{\mathbb{G}}$



Supporting Multiple Groups

Question 1: What if we want group-based access control, i.e., where consumers in the same group generate the same obfuscated name?

(One) Answer: Consumers in group \mathbb{G} share the encryption key $k_{\mathbb{G}}$

Question 2: How does a producer identify the correct decryption key for content?

(One) Answer: Include the group identifier in the payload of each interest, e.g.,

$$\text{ID}_{\mathbb{G}} = H(k_{\mathbb{G}}^P)$$



Supporting Multiple Groups (cont'd)

Question 3: How to prevent likability of multiple interests with the same ID_G ?

(One) Answer: Encrypt the identifiers using the publisher's public key pk^P

$$ID_G = \text{Enc}(pk^P, H(k_G^P))$$



Hash-based Obfuscation

$$N' = H(k, N)$$

where k is the same shared group key



Hash-based Obfuscation

$$N' = \mathbf{H}(k, N)$$

where k is the same shared group key

Introduces more state since a producer must be able to invert \mathbf{H} to discover N

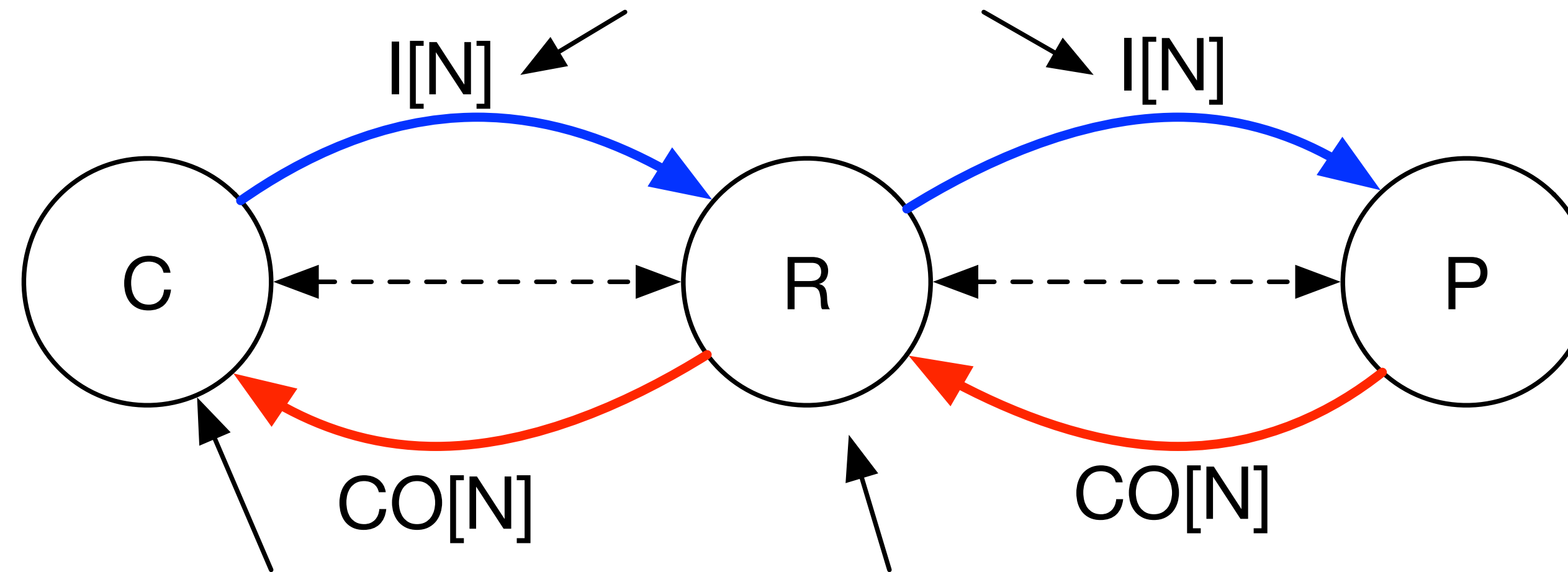


What about on-path attackers?...



Replay Attacks

1) issue interest I for IBAC-protected content with name N

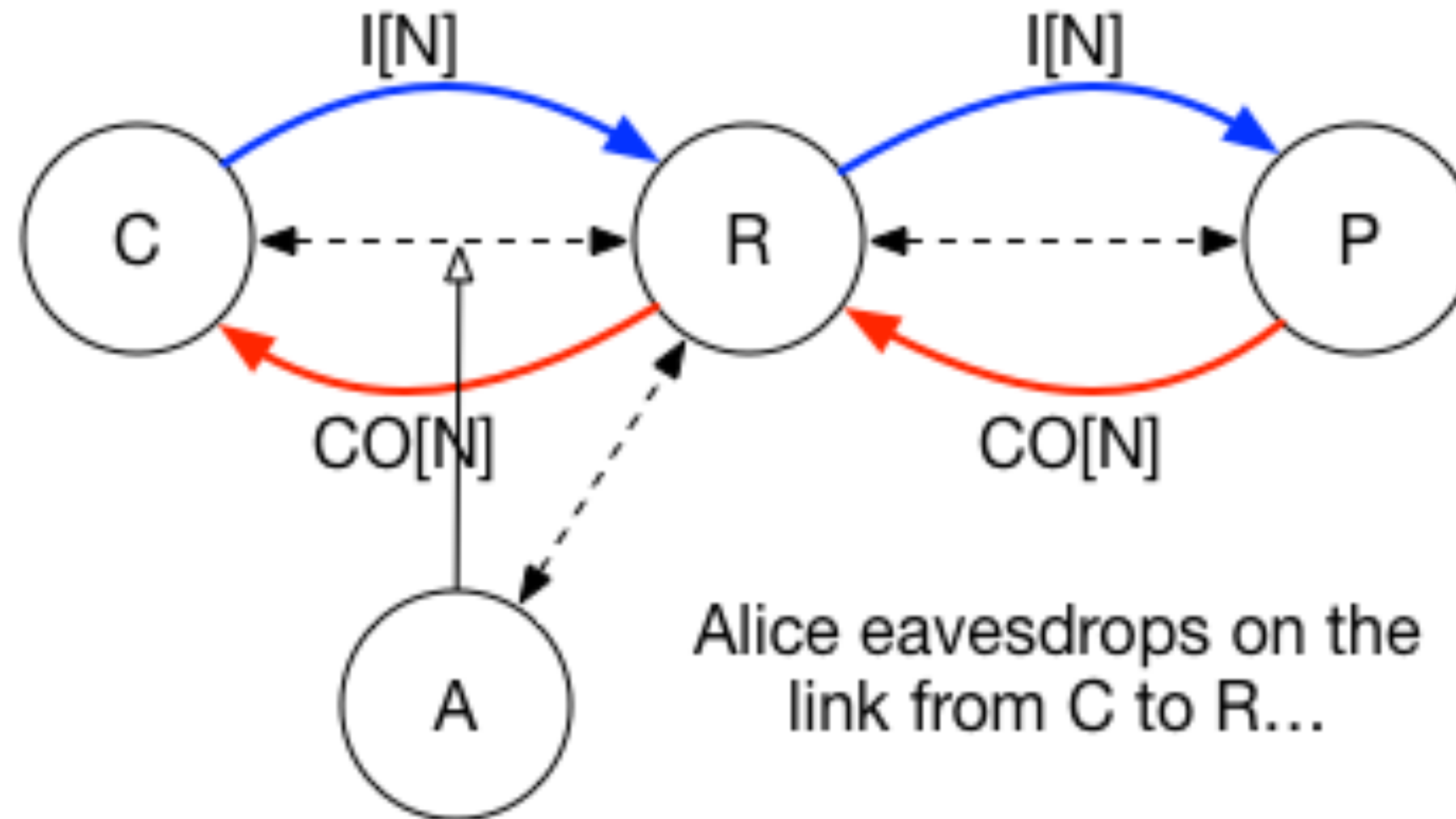


3) Consume content $CO[N]$

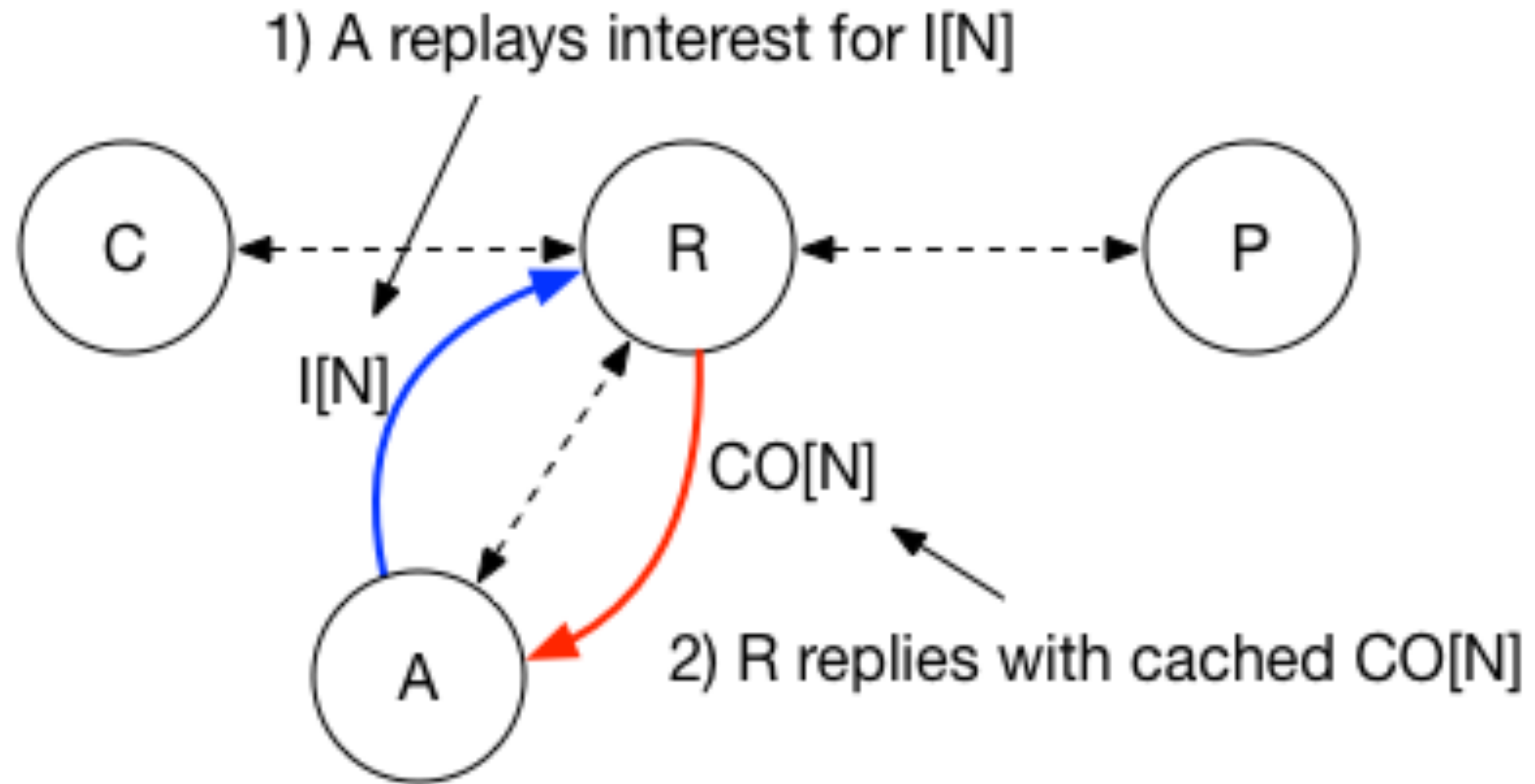
2) Cache IBAC-protected content $CO[N]$



Replay Attacks (cont'd)



Replay Attacks (cont'd)



Replay Attacks in Detail

Any (on-path) adversary can observe an obfuscated interest, replay it, and get the same content

Replay prevention:

- Nonces and timestamps help prevent replays
- ... in addition to consumer authentication information

$$\text{Payload} = \left(\text{ID}_{\text{G}}, r, t, \sigma = \text{Sign}_{sk_{\text{G}}^s} (N' || \text{ID}_{\text{G}} || r || t) \right)$$



Interest Authentication

Question: How can a router check if a given (cached) content object should be returned in response to an interest?

Answer: Verify an authenticator in interests (e.g., a digital signature)



Interest Authentication

Question: How can a router check if a given (cached) content object should be returned in response to an interest?

Answer: Verify an authenticator in interests (e.g., a digital signature)

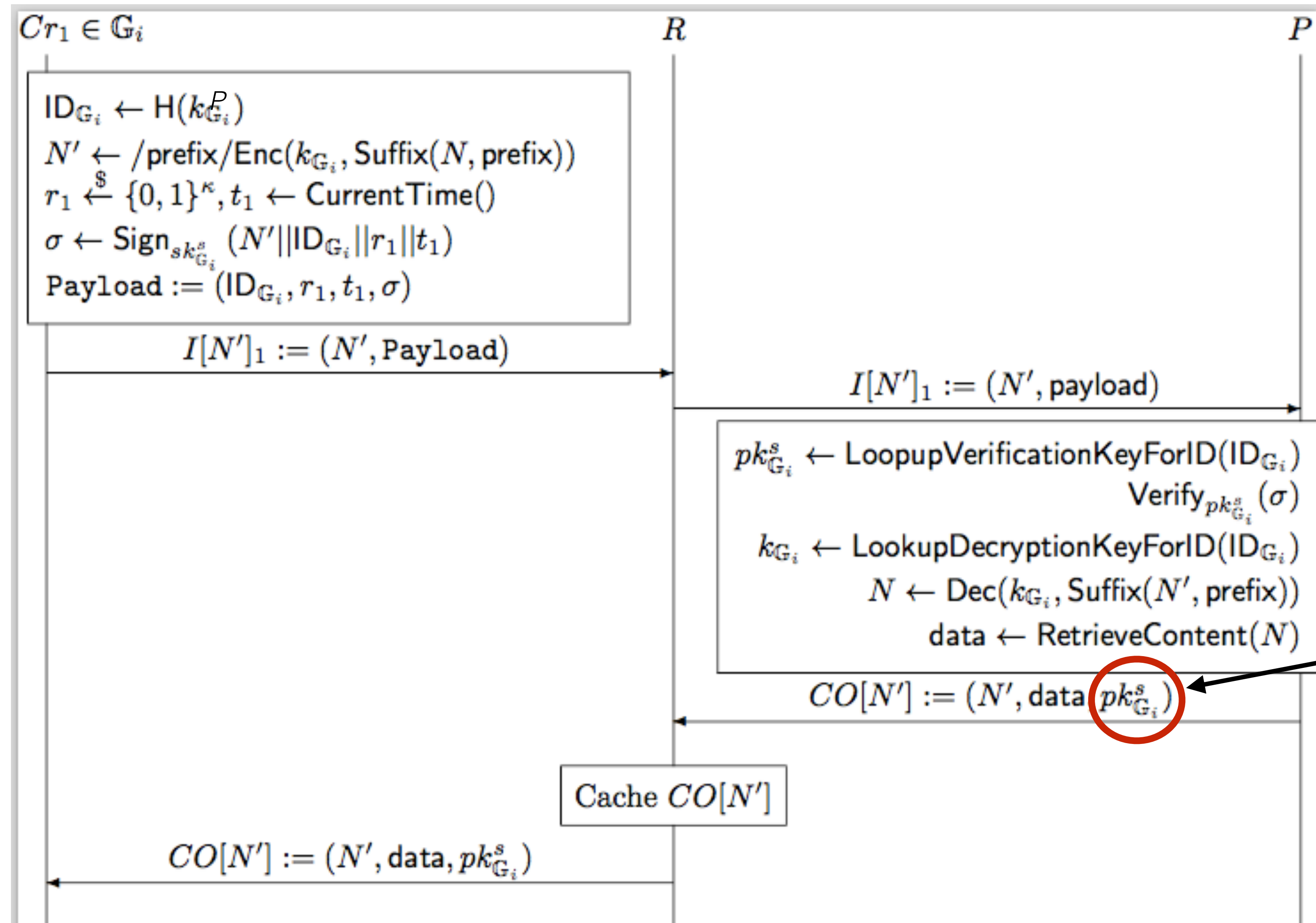
Question: How does a router know what key(s) to use for verification?

Answer: Follow the authorized content key binding (ACKB) rule:

ACKB: Cached content protected under IBAC must reflect the verification key associated with the authorization policy.



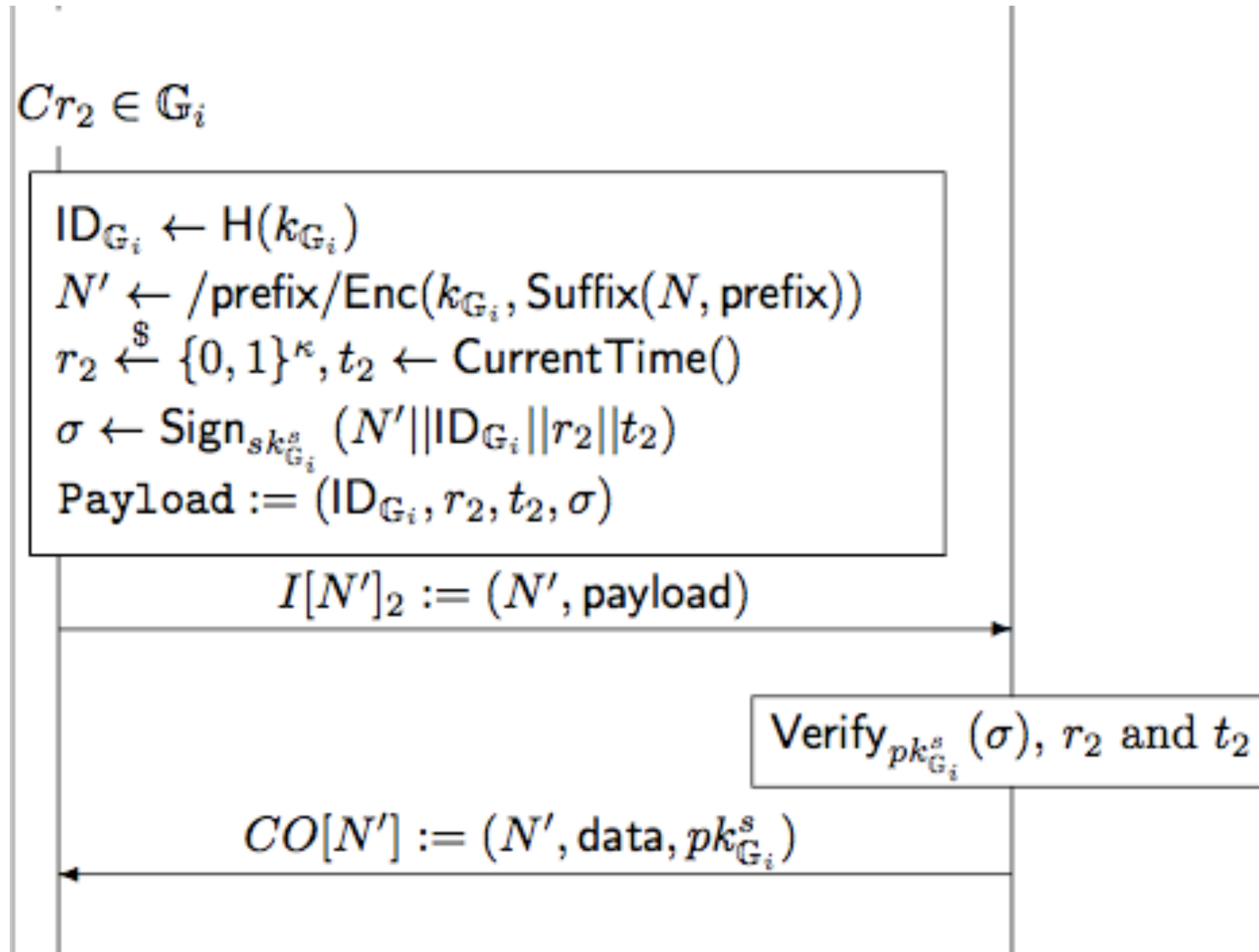
ACKB in Action (Part 1)



verification key



ACKB in Action (Part 2)



Verification Procedure

```
1: INPUT:  $I[N']$ , cached  $CO[N']$ ,  $B$ 
2:  $(ID_{G_i}, r, t, \sigma) := \text{Payload}$ 
3:  $(N', \cdot, pk_{G_i}^s) := CO[N']$ 
4: if  $B[N']$  contains  $r$  then
5:   Drop  $I[N']$ ; return Fail
6: else
7:   if Timestamp  $t$  is invalid then
8:     Drop  $I[N']$ ; return Fail
9:   else
10:    if  $\text{Verify}_{pk_{G_i}^s}(\sigma)$  then
11:       $B[N'] := B[N'] \cup r$ 
12:      return Pass
13:    else
14:      Drop  $I[N']$ ; return Fail
15:    end if
16:  end if
17: end if
```



Handling Policy Changes

- Policy changes include adding and removing users from groups
- Adding users to groups is easy (give them the right key)
- Removing users is hard:
 - Generate and distribute new group keys
 - Cached content may still exist in the network



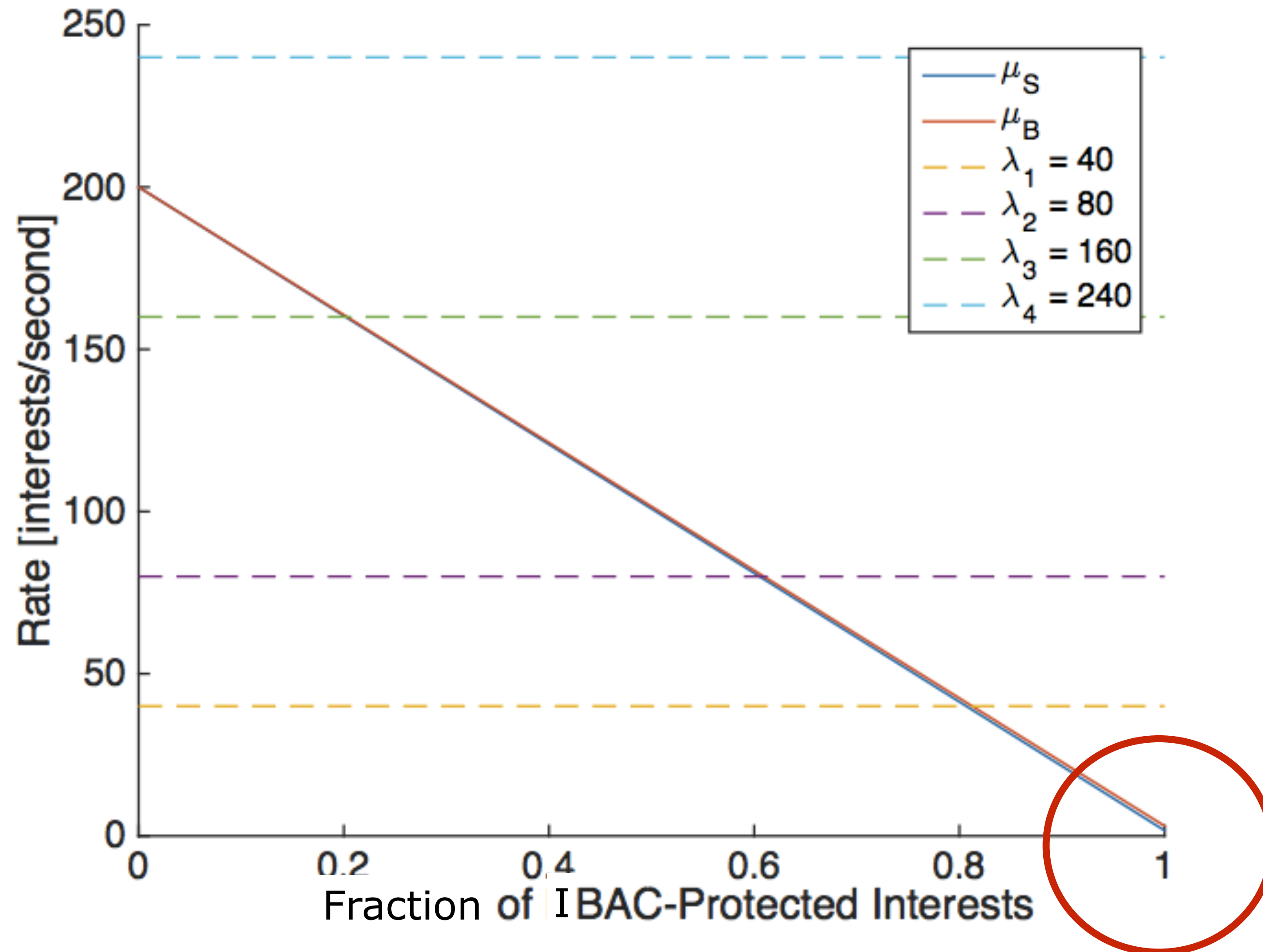
Experimental Assessment

- Without authorization checks, routers incur no added overhead
- With authorization checks, routers must:
 - Manage and verify nonces and timestamps
 - Verify interest signatures (!)

Key Size	Batch Size	Sig. Size	Indiv. Time	Batch Time	Improved
1024b	10	512KB	0.599s	0.322s	46%
1024b	10	8MB	0.888s	0.615s	30%
1024b	50	512KB	2.918s	1.579s	46%
1024b	50	8MB	4.315s	2.991s	30%
2048b	10	512KB	4.065s	2.207s	46%
2048b	10	8MB	4.104s	2.269s	45%
2048b	50	512KB	20.081s	11.029s	45%
2048b	50	8MB	21.301s	12.536s	41%
3072b	10	512KB	12.406s	6.789s	45%
3072b	10	8MB	12.804s	7.122s	44%
3072b	50	512KB	60.174s	32.877s	45%
3072b	50	8MB	64.347s	35.601s	45%



DoS Issues



Recommendations

- If replay attacks are not a concern, consumers use name obfuscation and include their group identity in interests.
- Otherwise, name obfuscation must be used and authorization information must be included in interests.
- If replay attacks are plausible but name privacy is not a concern, authorization information is sufficient.



Conclusion

1. Motivated content- and interest-based access control
2. Two ways to enforce IBAC
3. One way to handle replay attacks
4. Experimental assessment
5. Recommendations for using IBAC



Questions?...

