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 \( N \)

\[ U(N) \] set of consumers authorized to access content with name \( N \)

\[ \bar{U}(N) \] complement of the above

\[ G \] 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 $Cr \notin \mathbb{U}(N)$ then $Cr$ 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.


Content-Based AC in Pictures

Content Object

Header [/a/b/c]

Payload

ValidationSection
Content-Based AC in Pictures (cont’d)

[Diagram depicting the process flow of content object with header, payload, and validation section, and an encryption operation Enc(k, payload)]
Content-Based AC in Pictures (cont’d)

![Diagram of Content-Based AC in Pictures]

- **Header**: `/a/b/c`
- **Payload**: Encrypted with `Enc(k, payload)`
- **ValidationSection**: Unchanged
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

<table>
<thead>
<tr>
<th>Header [/a/b/c]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload</td>
</tr>
<tr>
<td>ValidationSection</td>
</tr>
</tbody>
</table>
Interest-Based AC in Pictures (cont’d)

Content Object
- Header \[/a/b/c\]
- Payload
- ValidationSection

Content Object
- Header \[/a/b/c\]
- Enc(k, name)
- Payload
- ValidationSection

obfuscated suffix
… why not do both?
Two Dimensions of AC

![Diagram showing two content objects with headers, payloads, and validation sections, each encrypted and obfuscated.]

- Header: `/a/b/c`
- Enc(k2, payload)
- Payload
- ValidationSection
- Header: `/a/b/c/k`
- Enc(k1, name)
- Obfuscated suffix

The diagram illustrates the two dimensions of AC, showing how content objects are encrypted and have obfuscated suffixes for security.
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 \( \text{Path}(Cr, P) \) be the set of all routers on the path between \( Cr \) and \( P \)

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

- 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 $G$ share the encryption key $k_G$. 
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 $G$ share the encryption key $k_G$

**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.,

$$ID_G = H(k_G^P)$$
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 $pk^P$

$$ID_G = \text{Enc}(pk^P, H(k^P_G))$$
Hash-based Obfuscation

\[ N' = H(k, N) \]

where \( k \) is the same shared group key
Hash-based Obfuscation

\[ N' = H(k, N) \]

where \( k \) is the same shared group key.

Introduces more state since a producer must be able to invert \( H \) to discover \( N \).
What about on-path attackers?...
Replay Attacks

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

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

3) Consume content $CO[N]$
Replay Attacks (cont’d)

Alice eavesdrops on the link from C to R...
Replay Attacks (cont’d)

1) A replays interest for I[N]

2) R replies with cached CO[N]
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

\[
\text{Payload} = \left( \text{ID}_G, r, t, \sigma = \text{Sign}_{\text{sk}_G} (N' || \text{ID}_G || r || t) \right)
\]
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)

$C_{r_1} \in G_i$

$ID_{G_i} \leftarrow H(k_{G_i}^P)$

$N' \leftarrow \text{prefix}/\text{Enc}(k_{G_i}, \text{Suffix}(N, \text{prefix}))$

$r_1 \leftarrow \{0, 1\}^e, t_1 \leftarrow \text{CurrentTime()}$

$\sigma \leftarrow \text{Sign}_{k_{G_i}^S}(N'||ID_{G_i}||r_1||t_1)$

$\text{Payload} := (ID_{G_i}, r_1, t_1, \sigma)$

$I[N'_1] := (N', \text{Payload})$

$pk_{G_i}^d \leftarrow \text{LookupVerificationKeyForID}(ID_{G_i})$

$\text{Verify}_{pk_{G_i}^d}(\sigma)$

$k_{G_i} \leftarrow \text{LookupDecryptionKeyForID}(ID_{G_i})$

$N \leftarrow \text{Dec}(k_{G_i}, \text{Suffix}(N', \text{prefix}))$

$\text{data} \leftarrow \text{RetrieveContent}(N)$

$CO[N'] := (N', \text{data}, pk_{G_i}^d)$

Cache $CO[N']$

verification key
ACKB in Action (Part 2)

$C_{r_2} \in G_i$

$ID_{G_i} \leftarrow H(k_{G_i})$

$N' \leftarrow \text{prefix}/\text{Enc}(k_{G_i}, \text{Suffix}(N, \text{prefix}))$

$r_2 \leftarrow \{0, 1\}^\kappa, t_2 \leftarrow \text{CurrentTime}()$

$\sigma \leftarrow \text{Sign}_{s_{k_{G_i}^s}}(N'||ID_{G_i}||r_2||t_2)$

PayLoad $:= (ID_{G_i}, r_2, t_2, \sigma)$

$I[N']_2 := (N', \text{payload})$

Verify$_{p_{k_{G_i}^s}}(\sigma), r_2 \text{ and } t_2$

$CO[N'] := (N', \text{data}, p_{k_{G_i}^s})$
Verification Procedure

1: INPUT: $I[N']$, cached $CO[N']$, B
2: $(ID_{G_i}, r, t, \sigma) := \text{Payload}$
3: $(N', \cdot, pk^u_{G_i}) := CO[N']$
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 Verify$_{pk^u_{G_i}} (\sigma)$ 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 (!).
DoS Issues

Fraction of IBAC-Protected Interests vs. Rate [interests/second]

- $\mu_s$
- $\mu_B$
- $\lambda_1 = 40$
- $\lambda_2 = 80$
- $\lambda_3 = 160$
- $\lambda_4 = 240$

Fraction of IBAC-Protected Interests: 0 to 1
Rate [interests/second]: 0 to 250

Circle at (1, 0)
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?…