Group Key Encryption

Christopher A. Wood
UCI and PARC
ICNRG Interim Meeting – IETF 96 – Berlin
July 17, 2016
Goal

• Specify how to encrypt replica-stored data under a common group key
  – Not how to manage that group key

• Defer access control management of group keys to named data to a higher layer in the stack
Data Layers

Root Manifest (RM)

Producer-signed LINK

Nameless Root Manifest (M) (tree node)

Hash-based LINKs

Nameless Content Object (tree leaf)
Encryption Layers

- Encrypted under consumer-producer session key
- Not encrypted
- Encrypted under broadcast encryption scheme #1

Diagram:
- Root Manifest (RM)
  - Producer-signed LINK
  - Nameless Root Manifest (M) (tree node)
    - Hash-based LINKs
    - Nameless Content Object (tree leaf)
Encryption Layers

- **Encrypted under consumer-producer session key**
  - Root Manifest (RM)
  - Not encrypted
  - Nameless Root Manifest (M) (tree node)
  - Encrypted under broadcast encryption scheme #1
  - Nameless Content Object (tree leaf)

- Producer-signed LINK
- Hash-based LINKs
- Served over a session
- Served over vanilla CCN
Message Types

- **Root Manifest (RM)**
  - Nameless Root Manifest (M) (tree node)
  - Nameless FLIC Manifest
  - Nameless Content Object (tree leaf)

Application-layer manifest that contains:
- Producer-signed LINK to M
- List of replica pointers (locators or LINKs)
- Encrypted content symmetric key
Nameless Content Object Construction

• Input:
  – Symmetric data encryption key DEK
  – Content object C

• Output:
  – C with payload encrypted under DEK
Nameless Manifest Construction

• Input:
  – Encrypted Content Object leaves C1, ..., Cn
  – Symmetric data encryption key DEK
  – Group key GK (KEK)
  – Producer private key SK
  – Data name N

• Output:
  – DEK encapsulated with GK
  – Nameless manifest tree with root M built on the leaves
  – Signed link that binds N to H(M)
Root Manifest Construction

• Input:
  – Encrypted DEK under GK
  – Producer-generated link for M
  – Data name N
  – ID of group key GK -- GK_{id}

• Output:
  – **Content object** with name N a body containing the signed link, encrypted DEK, and GK_{id}
(Full) Protocol

Consumer

Producer

Replica

Create session

Request data with name N

Return root manifest RM

Verify RM, parse M LINK, decrypt DEK

Resolve M

... ...

Manifest node (inner or leaf)

www.websequencediagrams.com
Obtaining Private Decryption Key
Lame Delegation

• Lame delegation is when RM points a namespace where M is not stored
• This occurs when the replica does not confirm the pointers in RM
Preventing Lame Delegation

In English:
• RM says M can be obtained at the replica
• The Replica Manifest says that M can be obtained under its namespace

H(M LINK) = H2
Replica pointer: Hash-based LINKs
Replica Manifest Construction

• Input:
  – M LINK
  – Replica names
  – Replica private key SK

• Output:
  – Replica manifest (signed by SK) with the hash of M LINK and list of replica names
Lame Delegation Variant

Create session
Request data with name N
Return root manifest RM

Verify RM, parse M LINK, decrypt DEK

Fetch Replica Manifest
Return Replica Manifest

Verify Replica Manifest

Resolve M
...
...
Manifest node (inner or leaf)
...
...
Simple Extensions

• Move data creation to the replica
• Producer and replica(s) exchange KEK
• Protocol:
  – Consumers ask replica(s) for N and get RM
  – Consumers ask replica(s) for encrypted data