Secure Content Dissemination in CCN

Christopher A. Wood and Ersin Uzun

parc® A Xerox Company

Real World Scenario

Bandwidth is expensive – Storage is cheaper

Question #1

How do we efficiently distribute content?

Answer

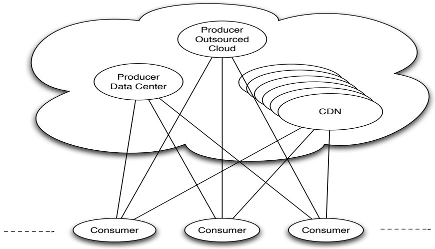
CCN / Content Delivery Networks

Question #2

How do we secure/individualize content?

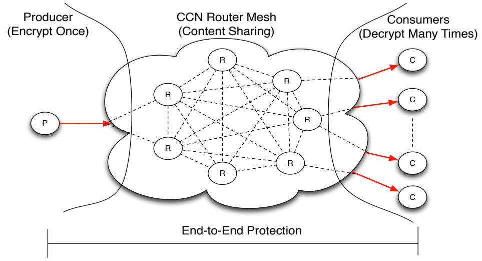
Answer

DRM technologies



Proxy Re-Encryption

- Transform ciphertext encrypted under one public key to one encrypted under another public key
- Enables secure sharing/re-sharing of content with re-encryption keys
- Encrypt content once, enable secure and scalable distribution from caches



DRM over CCN - Application Architecture

Full PRE-Based Approach

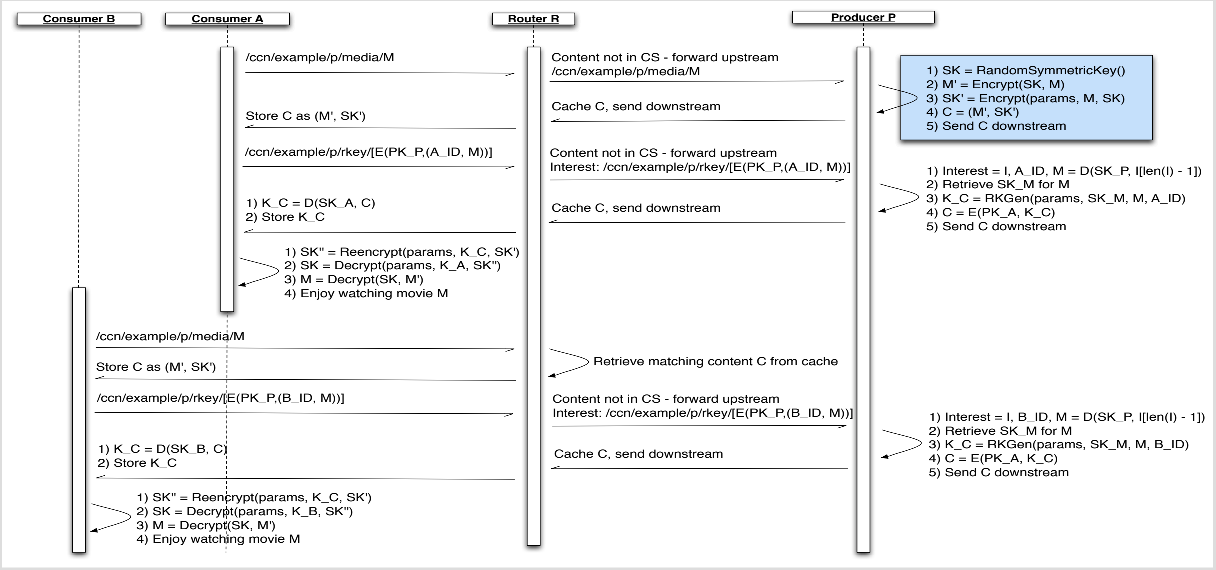
- Goal: Encrypt all content using PRE
- Problem: PRE operations are expensive
- Alternative: Hybrid symmetric and PRE encryption – still permits sharing across client devices without redundant key generation

One Time Setup

- Initiate individual user keys
- Distribute public parameters

Content Retrieval

- Request encrypted content (AES) and re-encryption key (PRE)
- Re-encrypt (PRE) and decrypt (AES) content



/ccn/example/p/media/M

Content not in CS - forward upstream
/ccn/example/p/media/M

1) SK = RandomSymmetricKey()
2) M' = Encrypt(SK, M)
3) SK' = Encrypt(params, M, SK)
4) C = (M', SK')
5) Send C downstream

Cache C, send downstream

Store C as (M', SK')

/ccn/example/p/rkey/[E(PK_P,(A_ID, M))]

Content not in CS - forward upstream
Interest: /ccn/example/p/rkey/[E(PK_P,(A_ID, M))]

1) Interest = I, A_ID, M = D(SK_P, I[len(I) - 1])
2) Retrieve SK_M for M
3) K_C = RKGen(params, SK_M, M, A_ID)
4) C = E(PK_A, K_C)
5) Send C downstream

Cache C, send downstream

1) K_C = D(SK_A, C)
2) Store K_C

1) SK'' = Reencrypt(params, K_C, SK')
2) SK = Decrypt(params, K_A, SK'')
3) M = Decrypt(SK, M')
4) Enjoy watching movie M

/ccn/example/p/media/M

Retrieve matching content C from cache

Store C as (M', SK')

/ccn/example/p/rkey/[E(PK_P,(B_ID, M))]

Content not in CS - forward upstream
Interest: /ccn/example/p/rkey/[E(PK_P,(B_ID, M))]

1) Interest = I, B_ID, M = D(SK_P, I[len(I) - 1])
2) Retrieve SK_M for M
3) K_C = RKGen(params, SK_M, M, B_ID)
4) C = E(PK_A, K_C)
5) Send C downstream

Cache C, send downstream

1) K_C = D(SK_B, C)
2) Store K_C

1) SK'' = Reencrypt(params, K_C, SK')
2) SK = Decrypt(params, K_B, SK'')
3) M = Decrypt(SK, M')
4) Enjoy watching movie M

Benefits

- Allows full utilization of caching in CCN
- End-to-end encryption from producer to application (key leakage is still possible in hybrid architecture)
- Strong content security and protection with individualized key encryption
- Fewer round-trip messages between producer/retailers

Future Work

- Use conditional (policy-based) PRE to add further content protection
- Integrate content fingerprinting into PRE process for traitor tracing
- Integrate PRE functionality into CCNx code base