Session-Based Content Distribution with CCNx-KE

Christopher Wood

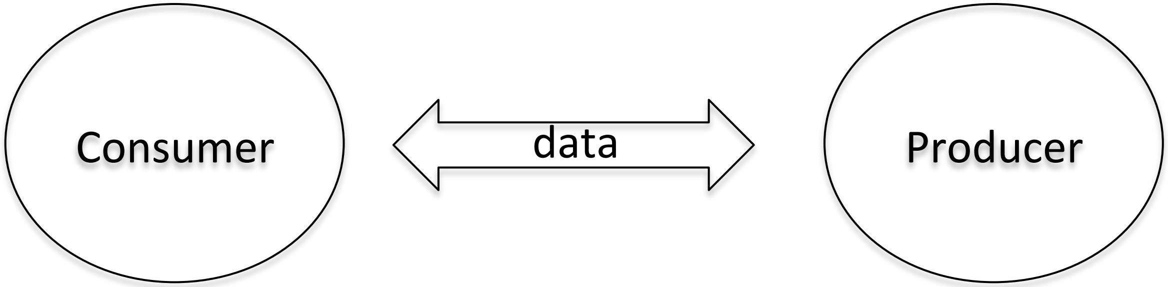
PARC, UCI

ICNRG 95 – Buenos Aires – 4/3/16

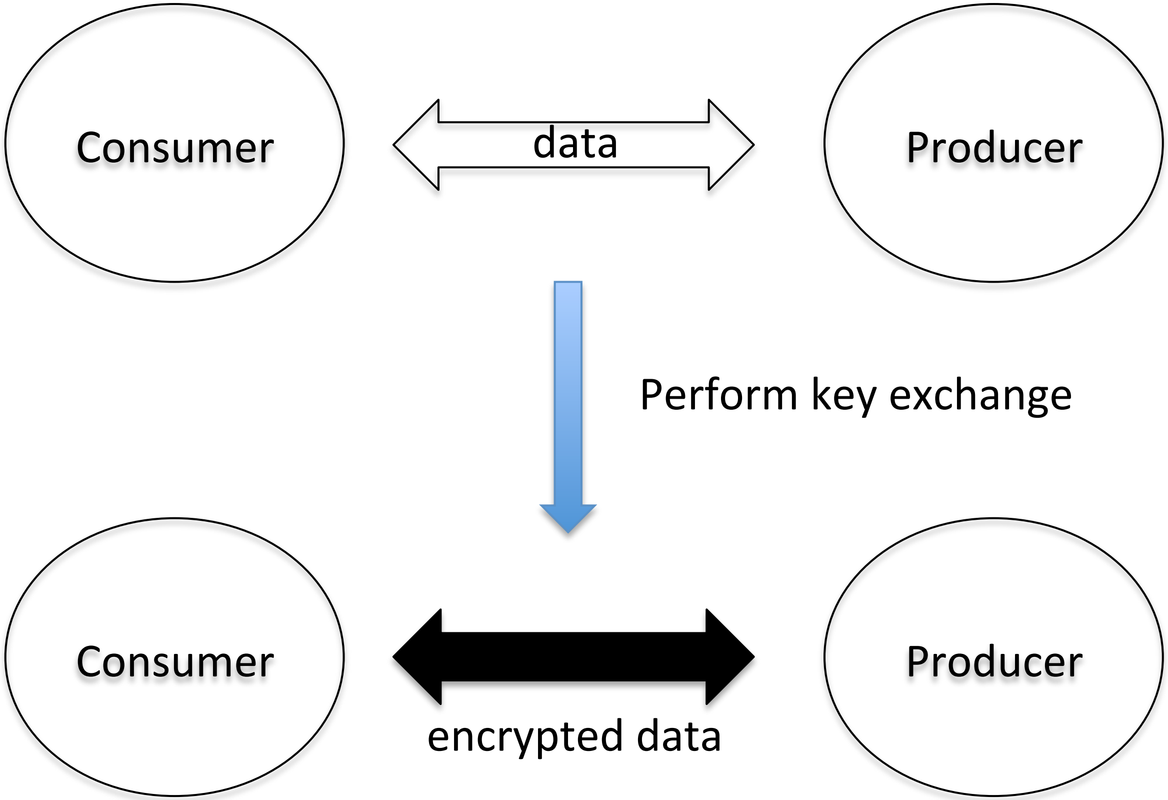
Session-Based Communication in CCN

- Problem:
  - A client and server (replica) want to establish a secure session in which all messages (interests and content objects) will be encrypted.
- Solution:
  - Use CCNx-KE – a TLS-like key exchange protocol tailored for CCN.
  - Clients authenticate the server (and vice versa) and the parties establish a shared forward-secure session key.
  - The session key is used to encrypt all subsequent traffic carrying application data.

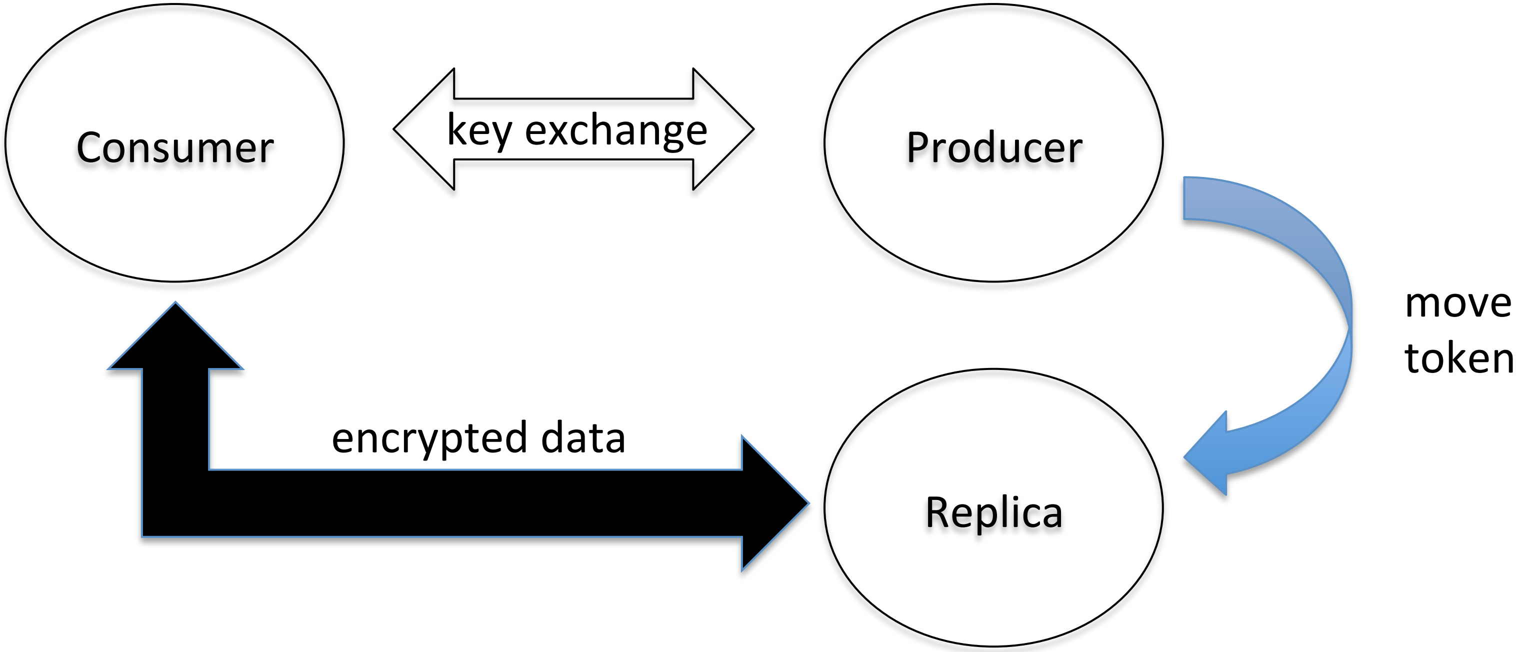
Standard CCN Session Communication



Standard CCN Session Communication



Ideal Session Relocation



CCNx-KE Features

- A consumer authenticates itself with a content producer and creates a forward-secure key and session.
- The content producer can serve content under that session or issue a **move token** to let another party serve content.
- Authentication and authorization are decoupled from data production
  - Benefits:
    - no private keys need to be shared between the server and replica
    - minimal information disclosure

Problems to Address

1. What is the trust relationship between the producer and the replica?
2. How is the move token transferred from the producer or the replica, or how is it created so that the replica can use it?

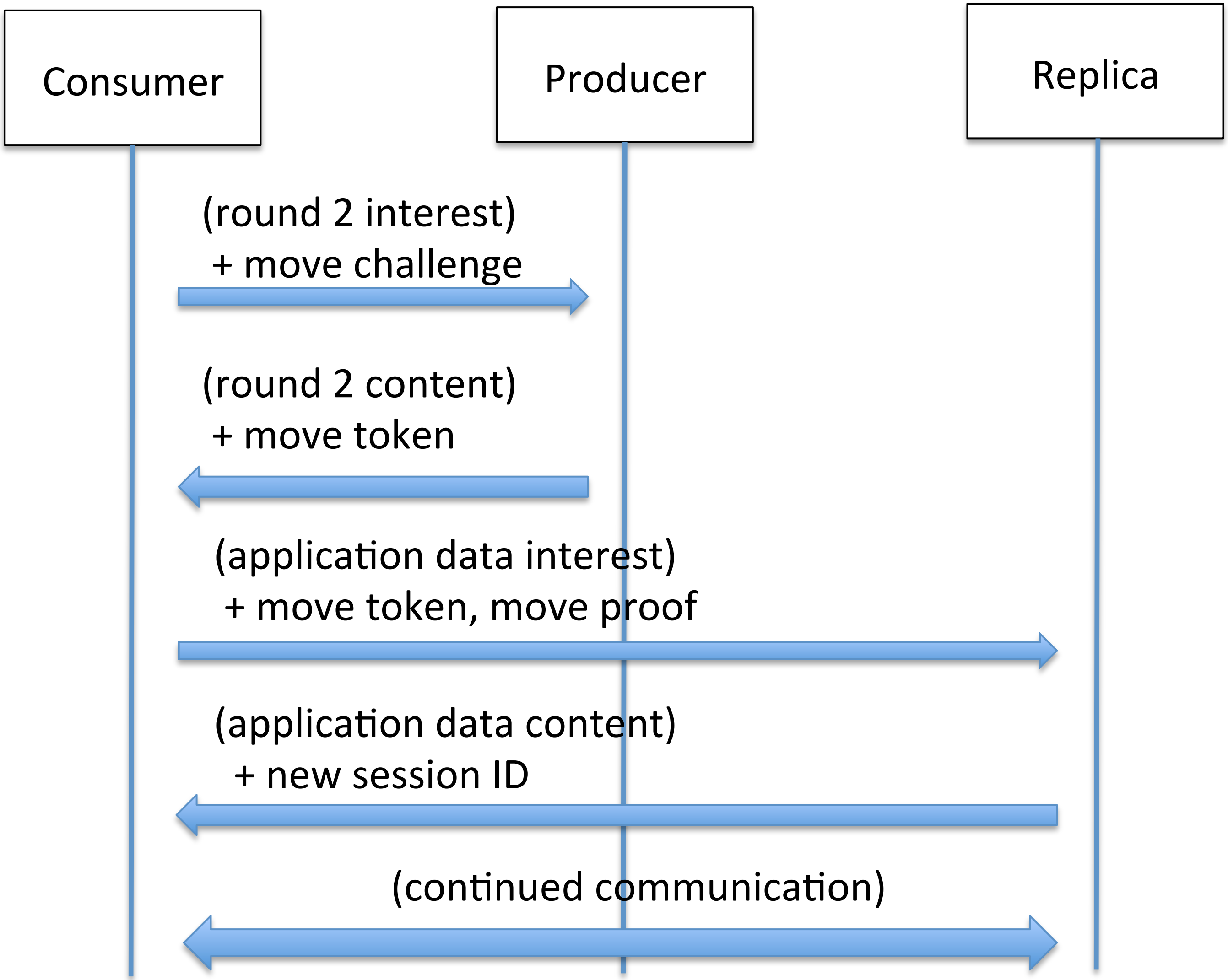
Trust

- The producer and replica have some relationship.
  - The producer pays for replica services.
  - A MNO distributes users to the best replica.
  - The authentication server passes the user to a load balancer (via a move token).
- The producer is capable of creating a secure channel between the replica.
- The producer and replica can create and share keys (and re-key) on a regular basis.

Move Token Goals

- A move token must enable the replica to decrypt interests and encrypt content responses
    - This requires the **traffic secret** established by CCNx-KE
- A consumer must **prove** that they fetched their move token from the producer

Move Token Usage



Move Token Construction

- Move challenge

  $Y = H(X)$, for some $X \leftarrow \{0,1\}^{128}$

- Move token

  $T = k_{ID} \,||\, Enc_k(Y \,||\, traffic\_secret)$

- Move proof

  X

Move Token Construction

- Move challenge

  $Y = H(X)$, for some $X \leftarrow \{0,1\}^{128}$

- Move token

  $T = k_{ID} \,||\, Enc_k(Y \,||\, traffic\_secret)$

- Move proof

  X

Replica check:

1. If k_{ID} not valid, drop
2. $Y \,||\, traffic\_secret = Dec_k(T)$
3. If $H(X) \mathrel{!=} Y$, drop

Properties

- k_{ID} is a key that's routinely refreshed between the producer and replica (e.g., on a daily basis).
- Replica work is minimized:
  - – no public-key crypto
  - – single symmetric decryption and hash computation
- Two round trips before data can be retrieved
  - – 1) Authenticate with the producer
  - – 2) Start a new session with the replica and get the first chunk of data

Summing Up

- CCNx-KE is used to separate authentication and authorization from the retrieval of actual application data.
- Producers can upload encrypted data to a replica that only authorized consumers can decrypt.
- The replica session is used as a form of "transport encryption."