A Tradeoff between Caching Efficiency and Data Protection for Video Services in CCN

2014. 2. 23.

Eunsang Cho*, Jongho Shin, Jaeyoung Choi, Ted “Taekyoung” Kwon, Yanghee Choi

escho@mmlab.snu.ac.kr
Network Convergence & Security Lab,
Seoul National University, Korea
BACKGROUND AND STATUS QUO
Background

• **Video content** is the one of major data sources with *massive volume*.

• **CCN** (Content-Centric Networking) is able to handle the video content well, thanks to *in-network caching*.
CCN In-network Caching

- First content request (Interest): from Bob to Alice
CCN In-network Caching

- First content delivery: from Alice to Bob
CCN In-network Caching

- Second content request (Interest): from Charlie to Alice

![Diagram showing CCN In-network Caching](image)

- Alice (Publisher)
- Charlie (Subscriber)
- Bob (Subscriber)
- Content Router
- Content Request (Interest)
- Content Data

Cached content from prev. content delivery
• Second content delivery: from cache to Charlie

CCN In-network Caching
With Encryption

- First content request (Interest): from Bob to Alice

Alice (Publisher) → Content Router → Content Request (Interest) → Encrypted Content Data for Bob → Bob (Subscriber) → Charlie (Subscriber)
With Encryption

• First content delivery: from Alice to Bob

Alice (Publisher) → Content Router → Bob (Subscriber) → Charlie (Subscriber)

D I D I D I D

Content Request (Interest)

Encrypted Content Data for Bob
With Encryption

• Second content request (Interest): from Charlie to Alice
With Encryption

- Second content delivery: from Alice to Charlie

Alice (Publisher)

Bob (Subscriber)

Charlie (Subscriber)

Content Router

Content Request (Interest)

Encrypted Content Data for Bob

Encrypted Content Data for Charlie

Cached content from prev. content delivery

Cache is ineffective!
Problem Definition

• **End-to-end data encryption** for each different content subscriber makes *caching ineffective*.

  – A novel video encryption scheme for CCN is required.
Objectives

• The objectives of this research are:
  – To develop a video encryption scheme which can utilize caching feature of CCN
  – To provide a practical approach for video content protection
  – To customize protection levels by video content provider’s requirements

→ To provide tradeoffs between data protection level, decodability of video, and cache effectiveness
Status Quo

- Transport Layer Security (TLS)

- Limitations
  - One-time validity of encrypted data
  - Ineffectiveness of in-network caching
Status Quo

- **Shared & symmetric key cryptography**

  - Key leakage problem
  - Untraceability of piracy

  **Diagram:**
  - Alice sends encrypted data to Bob.
  - Cache stores data for Bob and Charlie.
  - Cached data can be delivered to Charlie.
OUR PROPOSED SCHEME
Our Approach

• Access control with multiple symmetric keys
  – Distinct set of keys is assigned to each user
    • Tracing feature against key leakage problem (piracy)
  – Some keys can be shared among users
    • Subset of content can be shared by caching
Utilizing MPEG Video Structure

• MPEG video structure

A sample GOP sequence of MPEG video: GOP(12, 3)
Our Approach

• Video compression feature
  – From the structure of a MPEG video, some parts, such as I-frames are more important than others
    • Decrypting B- and P-frames requires I-frames
  – For higher cache utilization, less important parts can be left unencrypted

\[
\begin{align*}
  &k_1, k_2, \text{ and } k_4 \text{ are important parts. } \\
  &k_0: \text{ unencrypted}
\end{align*}
\]
Overview of the Framework

Publisher

Key Manager

Video Content

Encryptor

Unencrypted Video

CCN w/ Cache Storage

Key Distribution

Unencrypted Video

Encrypted Video

Subscriber

Decryptor

Resynch-rorizer

Video Player

Encrypted Video

Unencrypted Video
# Naming Model

## Video Content

<table>
<thead>
<tr>
<th>Scenes</th>
<th>Scene 1</th>
<th>Scene 2</th>
<th>Scene 3</th>
<th>Scene 4</th>
</tr>
</thead>
</table>

## Frames (not included in naming)

| I | B | B | P | B | B |

## Segments (=Packets)

<table>
<thead>
<tr>
<th>Seg. 1</th>
<th>Seg. 2</th>
<th>Seg. 3</th>
<th>Seg. 4</th>
<th>Seg. 5</th>
<th>Seg. 6</th>
<th>Seg. 7</th>
<th>Seg. 8</th>
<th>Seg. 9</th>
<th>Seg. 10</th>
</tr>
</thead>
</table>

- Encrypted
- Unencrypted
Operation Overview

1. Subscriber S requests her own set of keys for video.

2. Publisher P responds w/ multiple symmetric keys $\{k_1, k_2, k_3, \ldots, k_N\}$ and corresponding content names.

   $\begin{array}{ccccccccc}
   k_1 & k_0 & k_0 & k_2 & k_0 & k_0 & k_3 & k_0 & k_0 & k_1 & \ldots
   \end{array}$

   $k_0$: unencrypted

3. Subscriber S downloads packets of both encrypted and unencrypted video, the former of which are decrypted with symmetric keys in round-robin.
Do we need to encrypt all the segments of an I-frame?

- I-frames are larger than other frames in volume.
  - Usually an I-frame consists of multiple segments.
  - Encrypting a subset of segments may foil decoding the entire I-frame by adversary without proper keys.
Partial Encryption of I-Frames

• Not all the I-frame segments need to be encrypted.
  – Encrypting a subset of I-frame segments can lower PSNR significantly (of an adversary)
MODELLING AND EVALUATION
How Partial Encryption Affects the Performance?

Partial Encryption ($\rho$) → Avg. # of Differently Encrypted Copies for Each Segment ($E$) → Decodable Frame Rate ($Q$) → Cache Hit Probability ($\Phi_{hit}$)

- Partial Encryption ($\rho$)
- Decodable Frame Rate ($Q$)
- Avg. # of Differently Encrypted Copies for Each Segment ($E$)
- Cache Hit Probability ($\Phi_{hit}$)
Modelling Partial Encryption Impact on Decodable Frame Rate

- **Decodable Frame Rate** $Q$

\[ Q = \frac{N_{dec}}{N_{total}} = \frac{N_{dec-I} + N_{dec-P} + N_{dec-B}}{N_{total-I} + N_{total-P} + N_{total-B}} \]

- **Expected number of successfully decodable I-frames**
  - $p$: Encoded segment ratio of I-frame
  - Probability of the I-frame of a GOP to be successfully decoded ($C_I$: number of segments of an I-frame)

\[ S(I) = (1 - p)^{C_I} \]

- Expected number of successfully decodable I-frames

\[ N_{dec-I} = S(I) \times N_{GOP} = (1 - p)^{C_I} \times N_{GOP} \]
Modelling Partial Encryption Impact on Decodable Frame Rate

- Expected decodable frame rate $Q$

\[
Q = \frac{N_{\text{dec-I}} + N_{\text{dec-P}} + N_{\text{dec-B}}}{N_{\text{total-I}} + N_{\text{total-P}} + N_{\text{total-B}}}
\]

\[
= \frac{(1 - p)^{C_I} \cdot N_{\text{GOP}} + (1 - p)^{C_I} \cdot N_P \cdot N_{\text{GOP}} + \left(\left(\frac{N}{M} - 1\right) + (1 - p)^{C_I}\right) \cdot (1 - p)^{C_I} \cdot (M - 1) \cdot N_{\text{GOP}}}{N_{\text{total-I}} + N_{\text{total-P}} + N_{\text{total-B}}}
\]

\[
= \left\{1 + N_P + \left[\left(\frac{N}{M} - 1\right) + (1 - p)^{C_I}\right] \cdot (M - 1)\right\} \cdot (1 - p)^{C_I} \cdot N_{\text{GOP}}
\]

\[
= \frac{\frac{N}{M} + \left[\left(\frac{N}{M} - 1\right) + (1 - p)^{C_I}\right] \cdot (M - 1)\} \cdot (1 - p)^{C_I}}{N}
\]

$Q$ is inversely proportional to $p$. 
Evaluation of Partial Encryption

• Video Statistics
  – GOP(N=12, M=3)

<table>
<thead>
<tr>
<th>Video File</th>
<th>Foreman</th>
<th>Akiyo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of frames</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>I-frames</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Frames</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Total size of frames (Bytes)</td>
<td>435.643</td>
<td>312.528</td>
</tr>
<tr>
<td>P-frames</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Frames</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>Total size of frames (Bytes)</td>
<td>245.874</td>
<td>45.859</td>
</tr>
<tr>
<td>B-frames</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Frames</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Total size of frames (Bytes)</td>
<td>167.196</td>
<td>24.038</td>
</tr>
<tr>
<td>$C_I$ For 0.5K Packet</td>
<td>34.85144</td>
<td>25.00224</td>
</tr>
<tr>
<td>For 1K Packet</td>
<td>34.85144</td>
<td>25.00224</td>
</tr>
<tr>
<td>For 2K Packet</td>
<td>8.71286</td>
<td>6.25056</td>
</tr>
<tr>
<td>For 4K Packet</td>
<td>4.35643</td>
<td>3.12528</td>
</tr>
</tbody>
</table>

$C_I$ is the mean number of packets of an I-frame, which is used for previous model.

• Evaluation Method
  – Encoder/decoder
    • ffmpeg, libavcodec
  – Making pseudo encrypted file
    • Equal-length segments of I-frame is overwritten with meaningless 0x41 (‘A’) depending on probability $p$.
  – Quality Metric
    • PSNR
PSNR

- Peak Signal to Noise Ratio (PSNR) is the standard way to measure video fidelity.

\[
PSNR = 10 \log_{10}(\frac{c^2}{MSE})
\]

- $c$ is a maximum possible value of a pixel (constant)

- PSNR is measured in decibels (dB).
- Higher PSNR value means better quality.
(a) Measured PSNR (Y-YUV) of Foreman CIF, MPEG-4 H.264/AVC, GOP(12, 3)

(b) Expected Decodable Frame Ratio $Q$, GOP(12, 3)

(a) Original

(b) Best PSNR

(c) Worst PSNR
(a) Measured PSNR (Y-YUV) of Akiyo CIF, MPEG-4 H.264/AVC, GOP(12, 3)

(b) Expected Decodable Frame Ratio $Q$, GOP(12, 3)
Measured PSNR vs. Q

- 4K Packet: Linear Regression ($p = 0.9899$)
- 2K Packet: Linear Regression ($p = 0.9597$)
- 1K Packet: Linear Regression ($p = 0.8947$)
- 0.5K Packet: Linear Regression ($p = 0.7985$)
Modelling Cache Hit Probability

- Cache hit probability can be calculated on a single cache with a cache storage of \( m \) segments:
  - Hit probability of segment \( k \) \((k = 1, \ldots, K)\)
    \[
    \mathcal{P}^\text{hit}_k(m, E) = 1 - \pi_k^m = 1 - \frac{K' - m}{K'(q_k+1) - 1} \prod_{i=1}^{m-1} \left( \frac{K' - i}{K'(q_k+1) - 1 - i} \right)
    \]

- Hit probability of the whole \( K' \) segments
  \[
  \mathcal{P}^\text{hit}(m, E) = \sum_{i=1}^{K'} q_i \mathcal{P}^\text{hit}_i(m, E)
  \]

\( \mathcal{P}^\text{hit} \) decreases since \( K' \) is proportional to \( p \).

\[
K' = K \cdot E
\]

- \( K' \) is the total number of different segments including the encrypted segments
- \( K \) is the total number of segments before encryption
- \( E \) is an average number of differently encrypted segments for a given content
Modelling Cache Hit Probability

• # of Segments
  – Blu Ray Single Layer 25GB → 6.25M of 4KB segments

• Memory capacity (m)
  – Cisco ASR1000 Series Route Processors (RPs)
  – RP1: up to 4GB DRAM → 1M of 4KB segments

• Base values:
  – 6.25K segments (on the network)
  – 1K segments of memory capacity

• Two key distributions
  – Min keys: max overlapping keys
  – Max keys: min overlapping keys

• Other settings
  – S=u=100, s=3, I-frame ratio=0.3

u: # of subscribers (users)
s: # of keys given to a user
S: # of keys in total (managed by a publisher)
Finding Optimal Configurations

• Tradeoff model between the cache hit probability $P_{hit}$ and decodable frame ratio $Q$
  – Tradeoff function

\[
T(m, p, s, u, S) = \gamma \cdot P_{hit}^{\gamma}(m, p, s, u, S) + \frac{1}{Q(p) + \delta}, \quad \gamma, \delta > 0
\]

– Maximum cache hit probability by varying control parameter $p$

\[
\max_p T \quad \text{s.t.} \quad 0 \leq Q \leq \epsilon \quad 0 \leq p \leq 1 \quad 1 \leq u \leq S^s \quad Q, \ p \in \mathbb{R}, \ u \in \mathbb{Z}_+
\]
Numerical Results

- $\delta = 1.0, S = u = 100, s = 3$, I-frame ratio=0.3, $K = 6250, m = 100, GOP(12,3), C_I = 4.35643$
Conclusion

• Assuming MPEG video streams, we seek to achieve data protection while preserving the advantage of CCN’s in-network caching

• We present a CCN protection framework for video streaming services:
  – Key mechanism is the partial encryption
  – Tradeoff between the data protection and caching efficiency in CCN
END OF DOCUMENT