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

Eunsang Cho*, Jongho Shin†, Jaeyoung Choi‡, Ted "Taekyoung" Kwon* and Yanghee Choi*
*School of Computer Science and Engineering
Seoul National University, Seoul 151–744, Korea
Email: escho@mmlab.snu.ac.kr, {tkkwon, yhchoi}@snu.ac.kr
†Computer Science Department
Stanford University, Stanford, CA 94305–9025 USA
Email: jongho@cs.stanford.edu
‡Department of Electrical Engineering
KAIST, Daejeon 305–701, Korea
Email: kangggu13@gmail.com

***Abstract*—Content-centric networking (CCN) is a prominent future Internet architecture, which deals with content as a first class citizen. While CCN can achieve the efficient content delivery by name-based routing and in-network caching, it reveals many security issues to be investigated yet. We consider video streaming services as a representative example to investigate the tradeoff between data protection and caching efficiency in CCN. We seek to explore the relation among the cache hit ratio, key management overhead, and peak signal-to-noise ratio (PSNR) in MPEG video streaming in CCN. Moreover, we present a CCN security framework that provides privacy, access control, and user authentication. This paper also discusses both analytic models and simulation results of the varying degrees of data protection, decodability/quality of video, and cache effectiveness.**

I. Introduction

The traffic volume (and portion) of video content is expected to rise in the Internet [6, 16]. As video streaming over the Internet becomes the norm, *e.g.* YouTube service on the web, many stakeholders of video content business wish to protect their copyrights as they have been doing in the traditional movie/TV domains, *e.g.* Blu-ray disks and pay-per-view services. As a result, those stakeholders (such as Hulu, NetFlix, iTunes Store, and Amazon) have chosen to protect their online video content by encryptions, watermarking and so on.

The current growth rate of video traffic is likely to overwhelm the network capacity of most of network operators. Thus, the Internet community has been working for some solutions (*e.g.* CDN) in the current Internet, or has sought to fundamentally address the problem by proposing a new networking architecture (*e.g.* content-centric networking (CCN) [11]). In CCN, when an end user issues an *Interest* packet (*i.e.* a content request), it will be routed toward the content holder of the specified content name, and then the *Data* packet will be routed back to the end user along the path. The CCN router can then cache the content in its own storage (so-called in-network storage) in order to service later Interests for the same content, which results in efficient content delivery from a nearby storage to the end user. Thus, CCN's major strength comes from avoiding redundant transmissions of the same content due to the use of in-network caches (in routers). Also CCN inherently supports multicasting by making a CCN router maintain multiple forwarding interfaces (for a given content name) while sending only one interest toward the content holder. This multicasting entry is maintained in a new structure, which is called a Pending Interest Table (PIT). Note that the PIT also manages unicasting entries. On receipt of the corresponding data packet, the CCN router relays the data packet to the interface(s) recorded in the PIT entry, so that the redundant traffic of the same content is not transmitted. Overall, due to the usage of PITs and in-network caching, CCN can effectively reduce traffic and delivery time.

However, if the content is to be encrypted by each user's own key for confidentiality or data protection, the in-network caching in CCN will not be useful since each user has her own encrypted version of the same data. That is, cached data cannot be reused by others. Thus, as [4] suggested, providing data confidentiality (or data protection) while keeping the in-network caching effective is one of the open challenges in CCN. There are some candidate solutions for this problem, *e.g.* broadcast encryption [7], group signature [5], proxy re-encryption [2], and the access control enforcement delegation scheme for the PURSUIT project [8]. However, these schemes have drawbacks or limitations, such as the key sharing problem, heavy dependency on a specific entity, and high computational complexity (to be discussed later).

We seek to come up with a CCN security framework that provides data protection, access control, key management, and authentication, while retaining CCN benefits with low computational complexity. The main characteristics of the proposed framework are using symmetric key cryptographic systems and exploiting the MPEG video structure for video services. Under this framework, we substantiate how to trade

SENT '14, 23 February 2014, San Diego, CA, USA

http://dx.doi.org/10.14722/sent.2014.23008

off data protection and cache effectiveness in CCN.

This paper is organized as follows. Section II presents the problems, requirements and design considerations. Section III describes the proposed framework in terms of functional components: (i) key management and access control, (ii) video encryption and caching. Modelling and analysis are given in Section IV which focuses on the impact of partial encryption on decodable frame ratio and cache hit probability. Numerical results are presented in Section V. After discussing the related work in Section VI, we give concluding remarks in Section VII.

II. Objectives

A. Problem Definition

Video content is likely to account for increasingly more traffic volume in the future. CCN may help handle the video content properly by (i) in-network caching of popular content objects, and (ii) built-in support for multicasting. Note that the former is for video on demand (VoD) traffic, while the latter is for live streaming traffic. Now we focus on the caching aspect of CCN to reduce the VoD traffic.

In the case of the VoD traffic of popular content, the caching efficiency will be maximized if there is no encryption. However, for the purposes of data protection, the content publisher should encrypt the same video content with different keys for different users (for simplification, embedding randomness is not considered in this context). Data encryption with different keys makes caching mostly ineffective, which means the same segment of a video file will be differently generated/encrypted for different users, and hence the cached object of a user cannot be used for other users.

The objectives of this study are as follows. First, a novel video encryption mechanism is proposed to balance the trade-off between the caching effectiveness and data protection. Second, we present a general security framework for video delivery that can deal with not only the video encryption but also other functionalities such as user authentication, access control, and key management. Finally, we present mathematical models to analyze relations among the quality of video, degree of data protection, and cache utilization.

B. Design Considerations

The video encryption mechanism for CCN should be designed by considering the following requirements: (1) cache-friendly design, which means we seek to have as many same segments of video content as possible in caches; (2) access control to video content, which means only authorized users can play the video content properly; (3) low computation overhead to encrypt and/or decrypt video segments.

We will compare a few options of data protection for secure data transport in CCN. There are five choices depending on how keys are managed and which parts of a MPEG video stream is encrypted. Let us discuss the five options one by one.

1) Using TLS in CCN: Transport Layer Security (TLS) is a widely used in the current Internet. It establishes a secure transport channel with a shared session key between two hosts. One session key is generated for each session, and thus its lifetime is limited to the session duration. The first problem of using TLS in CCN is that the valid lifetime of a key of a session is much shorter than that of a cached object. Also, the trust of TLS is designed based on a synchronized session between two hosts. Meanwhile, a content object can be cached anywhere and retrieved asynchronously. That means if we use TLS for encrypting a content object in CCN, it will be valid only for the first client who issues the Interest and the cached copy will not be useful for other clients.

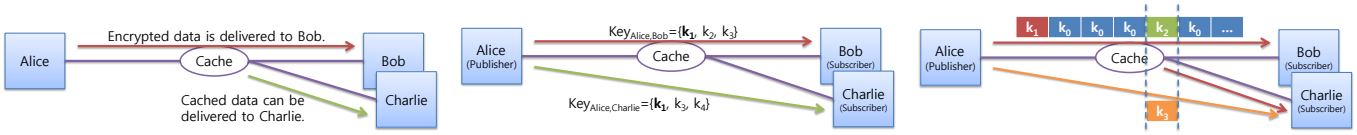
Thus, the current TCP-IP based approach for secure transport, such as TLS, cannot make use of cached storage due to its one-time validity. Therefore a novel secure transport mechanism for CCN needs to be designed.

2) Using Symmetric Key Cryptography: We assume that the public key cryptography (PKC) is used in CCN for authenticating content objects as discussed in [11]. However, content encryption using PKC is not a good idea due to its high computational cost. Instead, using symmetric key cryptography is desirable for low computational cost and for efficient caching. For example, we can cache the encrypted content during the validity of the key. The encryption scheme can be AES as a block cipher and SHA-256 as a HMAC. To make CCN's in-network caching effective, the validity of a key should be set to much longer duration (than a single transmission), for reuse of cached objects as shown in Figure 1(a). However, if a key leakage occurs, all the cached objects should be removed; otherwise, unauthorized users can access the objects.

3) Access Control with Multiple Symmetric Keys: Access control with the symmetric key cryptography is well defined in TLS. First, authentication of both parties— the publisher and the client—can be done with the PKC mechanism[1]. We assume that if a client is authorized (*i.e.* has a valid downloading right) for the content, then she obtains multiple symmetric keys as shown in Figure 1(b). She will rotate the received multiple symmetric keys to decrypt successive segments, and the rotation is repeated in rounds (in this illustration, a round consists of 3 segments). In the illustration, Bob has three keys *k1, k2, k3* to decrypt the video segments, and Charlie has three keys *k1, k3, k4* to decrypt the video segments. Thus, in this case, one out of three cached segments can be reused. Note that different sequences of keys (to decrypt successive segments) are assigned to different users. One of the reasons is that we can trace the user who is responsible for the key leakage.

4) MPEG Video Structure: The importance of data of a MPEG video stream is different part by part. To facilitate high utilization of caching, we should distinguish important parts from the others. The MPEG video basically consists of three kinds of pictures: Intrapictures (I), Predicted pictures (P) and Interpolated pictures (B—for bidirectional prediction) [14]. I-pictures (or I-frames) are independent of other type of pictures and therefore they are also called the reference pictures. P-pictures (or P-frames) are dependent on the preceding I- or P-

[1]For CCN security, publisher authentication is well detailed in [11], but client authentication is not discussed.



(a) Using Symmetric Key Cryptography (b) Access Control with Multiple Symmetric Keys; (c) Cache-friendly Design; k_0 means unencrypted. k_n indicates a symmetric key.

Fig. 1. Diagrams for Design Considerations

pictures. B-pictures (or B-frames) are dependent on both the preceding and the following I- or P-pictures. Therefore the importance of I-pictures is higher than that of the others, and also the errors in I-pictures will be propagated to P- and B-pictures due to dependency between picture types.

A MPEG video stream has multiple groups of (successive) pictures (GOPs). A GOP is defined as the successive pictures between two consecutive I-pictures. The GOP pattern is described by two parameters $GOP(N, M)$ [13], where N defines the distance between two consecutive I-pictures and M defines the distance between I-to-P or P-to-P pictures. While the two terms—*frame* to *picture* are interchangeable in the MPEG technologies. We usually use I-, P-, and B-frames throughout the paper.

5) Cache-friendly Design: Observing the structure of the MPEG video compression (detailed in Section II-B4), we note that I-frames are much more important than P- and B-frames. In other words, if I-frames are missing, the corresponding intervals are not played even if the relevant P- and B-frames are received/decrypted. Meanwhile, missing P- and B-frames have a relatively small effect since I-frames can be decoded independently. Also, not all I-frames need to be encrypted. Even if an adversary can decode some I-frames, the quality of video streaming can be intolerable depending on the ratio of encrypted I-frames. For the higher cache utilization, it is desirable to encrypt as few frames as possible as illustrated in Figure 1(c), to be detailed in the next section.

C. Security Model

Data confidentiality is usually ensured by proper encryption. Our focus in this paper is a VoD service which guarantees a required level of protection to limit unauthorized viewing. For example, satellite or cable TV services scramble video channels, depending on the required level of protection for commercial video contents.

The goals of an adversary are to eavesdrop video content and to play the video with the desired quality without proper rights. Such adversaries can be internal or external. An internal adversary is a user who watches a video legally and tries to reuse the keys to play another video. In other words, it is a user who does not possess the entire (authorized) key sequence for a video. An external adversary has no keys to play a video content. In either case, adversaries are assumed to compromise only a subset of network entities.

Our model allows the adversary to compromise routers (and caches) and to tap links. Using the two actions, the adversary can list up the series of segment names and receive the corresponding segments. Hence, the adversary can download the full or parts of video segments. Against this threat, our scheme lowers the quality of video below the tolerable level, and hence the adversary does not take advantages of eavesdropping the entire traffic.

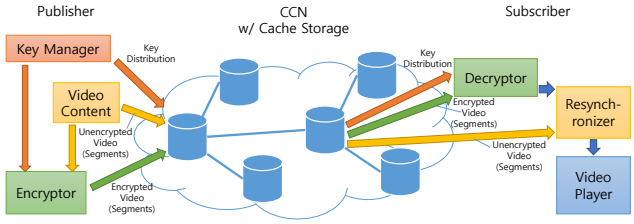


Fig. 2. Overview of the Proposed Security Framework

III. Proposed Framework

Our goal is to provide a security framework for CCN that can support the partial video encryption mechanism as depicted in Figure 2. Key management and encryption/decryption components are shown in the security framework to explain their roles in partial video encryption. Thus, we explain the security framework by detailing two key components: (1) key management and access control, (2) video encryption and caching.

A. Key Management and Access Control

CCN has authentication mechanisms for publishers and for content objects based on the PKC in the network layer. Also, access control models are deemed application-dependent [11]. However, subscriber authentication in CCN is not easy because there is no explicit identifier of a subscriber. Even if the publisher wishes to perform access control or encryption, it may not always be possible since Interests are often aggregated (*e.g.* multicasting) or cache hits will deliver content objects without the intervention of the publisher. Such mechanisms of CCN prevent the publisher from distinguishing (and authorizing) subscribers.

To overcome this problem, a subscriber-specific and unique content name can be used. In the Interest, the content name, her name (*e.g.* her ID in the publisher's domain or email address)), a nonce[2], and the digital signature (by the subscriber's private key) are encrypted by the publisher's public key, except for the publisher name (which is user for name-based routing). For example, `ccn:/pub.com/mov.mpg/id/nonce/sig` is encrypted to `ccn:/pub.com/3?dFienlFowhef...3ivn`. The publisher name will be unique in the network, and the Interest will be forwarded directly to the publisher without the Interest aggregation or cache hit. Note that the privacy of subscriber's requests is somewhat protected by revealing only the publisher names. Then the publisher can identify the

[2]A nonce can be placed either in the name field or in the nonce field of the Interest packet, depending on the structure of the Interest packet [11].

subscriber with her identity information, and the subscriber can be authenticated with the nonce and the signature. The subscriber's identity information is also hidden except for publishers. However, only using such subscriber-specific content names for subscriber authentication still has some drawbacks. For instance, it incurs PKC-based encryption and decryption for every Interest, whose computational cost is expensive.

Our approach is to adopt the symmetric key cryptography with multiple shared keys and to exchange a list of content names for the successive Interests in order to reduce the drawbacks of the above subscriber authentication. When a subscriber contacts the publisher for the first time, an Interest with a subscriber-specific and unique content name is issued. After the subscriber authentication, the publisher replies with multiple shared keys and a list of content names that are the segments (of the whole content). From then on, the subscriber can request the segments, and the delivered segments will be decrypted with the shared keys in a round robin fashion. This approach is taken by considering the requirements in Sections II-B1, II-B2, and II-B3. If the publisher wishes to provide the privacy of the segment names (for subscribers), the segment names can be hashed optionally.

In addition to the low computational complexity and less linkability between the subscriber and the requested content, this approach also has an advantage against the key leakage problem. As multiple shared keys are given to a subscriber, this subscriber can be distinguished (from another) by her unique sequence of the keys.

The symmetric shared key system has a management problem of key validity. If a key has its validity period and it is expired, the key should be revoked. The publisher, as a key manager, manages the key validity by changing the key pool gradually, which means keys are sorted in order of expiration times and the key manager replaces the oldest key with a new one when it has expired. If the expired key is revoked, the encrypted segments with new keys, provided by the publisher, cannot be decrypted with old keys. To keep caching effective during a relatively long term, the revocation is not an easy issue, to be discussed in Section III-B.

B. Video Encryption and Caching

In Section III-A, multiple shared keys are introduced for subscriber authentication. These keys are used for encryption of video segments and each key corresponds to a segment (the sequence of keys are repeated in rounds) as discussed in Section II-B3. The sequence of keys is uniquely determined for each user by the publisher. Therefore there is no explicit linkage with the shared keys and the name of the content segments.

In order to increase the caching probability, exploiting the characteristics of the MPEG video structure will be helpful as discussed in Section II-B4. By making the partial encryption exploit the MPEG structure, we can find the tradeoff between caching efficiency and data protection (or confidentiality). If we want to increase the cache hit probability, we just need to encrypt less frames. This approach is taken due to the design considerations in Section II-B5.

With the revocation of expired keys, cached segments encrypted with the expired keys should be removed. In a naive approach, a cached segment may eventually be replaced by another segment by a cache replacement policy (*e.g.* LRU or LFU). If the cache replacement is not frequent, a stale segment may stay in a cache for a substantial duration. Against this case, we add an expiration time field in the header of a content object such as *FreshnessSeconds* field of CCNx packet, so that the cache manager can remove the stale segment timely.

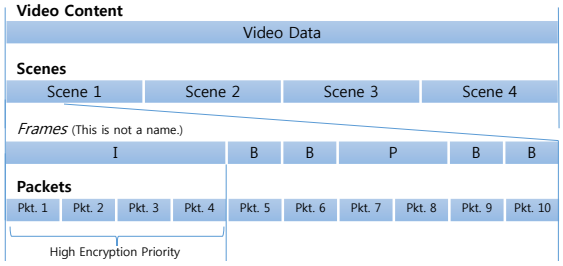


Fig. 3. An Overview of Naming Model

To achieve the high cache hit probability, we can consider multiple choices in designing a key distribution policy. If a sequence of keys is selected by the uniform distribution, it may lower the cache hit ratio on average. On the other end, if every key sequence is generated with the objective of maximizing overlapping keys with the prior sequences (that have been assigned), it will help increase the cache hit ratio. For example, suppose a key sequence for a user is represented by a string, (k1, k2, k3, k4), and the next key sequence for another user is (k1, k2, k3, k5), then 3/4 of encrypted segments can be shared between the two users. However, the latter key distribution is more vulnerable to the key leakage. When a key sequence is leaked, a large portion of the encrypted segments may be decrypted by the unauthorized access. Hence, choosing a proper key distribution policy considering various requirements is left for future work.

IV. Modelling and Analysis

In this section, we provide a mathematical model of the partial encryption, which trades off the data protection and caching effectiveness. Based on the model, we formulate the optimization problem which represents the tradeoff between these two metrics as control parameters. For brevity, the notation used in this model is summarized in Table I.

A. Naming Model

For convenience, a naming model for the content objects is shown in Figure 3. When a subscriber wishes to download a video file, her Interest is sent to the corresponding publisher. The publisher first responds with a list of names of the video partitions, called *Scenes*. Next, the subscriber sends an Interest with a name of a scene, and then a list of names of the *Packets* for the scene is received. The packet corresponds to a video segment in this paper and it is contained in a single transmission unit. For sake of clarity, a video segment is the original video data fragment of an I-frame, while a packet may refer to the encrypted version of the segment. Of course a packet may be unencrypted depending on the encryption policy and mechanism.

From Figure 3, the (video) frames are mentioned. The video frames do not have corresponding names in the view

TABLE I. NOTATION

Symbol	Explanation
K	Total number of packets
E	Average number of differently encrypted packets for a given segment
$K^{'}$	Total number of different packets including the encrypted packets ($K^{'} = K \cdot E$)
S	Number of multiple symmetric keys in total (managed by a publisher)
s	Number of multiple symmetric keys given to a subscriber
u	Number of subscribers (users)
p	Ratio of the encoded I-frame packets to the total number of I-frame packets
$f(s, u, S)$	A function that represents how many encrypted versions exist for each encrypted packet
N, M	I-to-I , I-to-P (or P-to-P) frame distance
GOP(N,M)	The GOP pattern
Q	Decodable frame ratio, $0 \leq Q \leq 1$
$N_{total-I}$, $N_{total-P}$, $N_{total-B}$	The total number of I, P, B-frames, respectively
N_{total}	Total number of frames, $N_{total} = N_{total-I} + N_{total-P} + N_{total-B}$
N_{dec-I}, N_{dec-P}, N_{dec-B}	The expected number of successfully decodable I, P, B-frames, respectively
N_{dec}	The expected number of Successfully decoded frames, $N_{dec} = N_{dec-I} + N_{dec-P} + N_{dec-B}$
C_I, C_P, C_B	The mean number of packets of an I, P, B-frame in a GOP sequence, respectively
$\mathbb{P}(I)$	The probability that the I-frame within a GOP to be successfully decoded
$\mathbb{P}(P_i)$, $\mathbb{P}(B_i)$	The probability that the i-th P, B-frame to be decodable within a GOP, respectively
N_{GOP}	The total number of I-frames (The total number of GOPs)
N_P, N_B	The total number of P, B-frames in a GOP, respectively
C	The average ratio of cacheable packets in a GOP
m	The size of cache storage
λ	The mean arrival rate of requests
q_k	The Zipf popularity distribution among K packets, $k = 1, 2, ..., K$
π^i	The stationary probability of finding a given packet in the cache in state i ($i = 1, 2, ..., m + 1$)
P_k^{hit}, P^{hit}	The cache hit probability for a given packet k, for all the packets, respectively
$T(m, p, s, u, S)$	The tradeoff function

of this paper, however they have important roles for caching efficiency.

B. Operation of Partial Encryption

For the video encryption, we assume that both the publisher and the subscriber use multiple symmetric keys. The operation of partial encryption is as follows. First, a subscriber requests keys for downloading of the segments (or packets) to the publisher. Then, the publisher responds with multiple symmetric keys $\{k_1, k_2, k_3, \cdots, k_s\}$ and the corresponding content names (that can identify packets) to the subscriber. Finally, the subscriber downloads the encrypted packets and decrypts them with multiple symmetric keys. Note that there are also unencrypted packets. If the packet encryption probability, p, is 0.2, the first packet out of every 5 packets is encrypted. To maximize the cache effectiveness, the ratio of encryption should be minimized. However, an adversary who receives both the encrypted and unencrypted packets should experience sufficiently poor video quality since she has no symmetric keys. Therefore, the impact of the partial encryption on video quality should be quantified.

C. Model

1) Modeling Partial Encryption Impact on Video Quality: In this subsection, we discuss the impact of the partial encryption among the whole packets of a video file (which actually applies only to I-frames) on the percentage of the decodable frames. A decodable frame means all the packets of the frame is successfully decoded by an adversary. (Obviously, an authorized user with the keys will decode all the frames.) The authors in [13] proposed the model of packet loss impacts on video quality. In our work, the packet loss model can be modified to reflect the partial encryption because the encrypted packets can be seen as the corrupted packets, which result in undecodable frames to an adversary. The model in [13] assumes that the packet loss rate is constant for the whole video frames. On the contrary, our partial encryption focuses on the important frames such as I-frames. Therefore, our model is the specialized form of their model. The model in [13] also considers the error propagation of undecodable frames due to the interdependency of MPEG video frames. On the other hand, our model assumes that the decoder at the user side discards a corrupted video frame and repeats the previous frame.

a) Video Quality Metric: We use an objective evaluation metric to assess video quality, known as Decodable Frame Ratio Q, as presented in [13]. Q is defined as the ratio of the number of successfully decoded frames (N_{dec}) to the total number of frames (N_{total}) of a video file; $Q = \frac{N_{dec}}{N_{total}}$

b) The Expected Number of Successfully Decodable I-frames (N_{dec-I}): We assume that there is no chance of packet loss from other causes. Then, the probability that the I-frame of in a GOP to be successfully decoded (by an adversary) is $\mathbb{P}(I) = (1-p)^{C_I}$, where $(1-p)$ represents the probability for an I-frame packet to be successfully received/decoded. Then, the expected number of successfully decodable I-frames for the whole video file can be obtained as

$$N_{dec-I} = \mathbb{P}(I) \cdot N_{GOP} = (1-p)^{C_I} \cdot N_{GOP}.$$

Clearly, there is one I-frame in each GOP.

c) The Expected Number of Successfully Decodable P-frames (N_{dec-P}): Since we assume that there are no packet

losses, the decodability of the first P-frame is affected only by the preceding I-frame. Hence, the probability of the first P-frame to be decodable within a GOP is $\mathbb{P}(P_1) = \mathbb{P}(I) = (1-p)^{C_I}$. The other P-frames are also affected by the previous I-frame only. As to the other P-frames in a GOP sequence, we can calculate $\mathbb{P}(P_i) = \mathbb{P}(P_1) = (1-p)^{C_I}$, where $i = 2, 3, ..., N_P$. Thus, the expected number of successfully decodable P-frames of N_{GOP} GOPs can be obtained as

$$N_{dec-P} = (1-p)^{C_I} \cdot N_P \cdot N_{GOP}.$$

d) The Expected Number of Successfully Decodable B-frames (N_{dec-B}): Within a GOP, B-frames are successfully decodable only if the preceding and succeeding I- or P-frames are both decodable. Thus, there are two types of B-frames in view of dependency on I-frames: (i) indirectly dependent on the preceding I-frame, (ii) indirectly dependent on two consecutive I-frames which are preceding and succeeding ones. In a GOP, the last successive B-frames (i.e., the last B-group) are the latter case, and the other B-frames/groups are the former case. Therefore, the number of successfully decodable B-frames for the former case is given by

$$\mathbb{P}(B_j) = \mathbb{P}(I) = (1-p)^{C_I},\ j = 1, 2, ..., (N/M) - 1.$$

where $\frac{N}{M}$ is the number of B-groups in a GOP when GOP(N, M) is given. The last B-group is indirectly dependent on two successive I-frames. Hence, it can be expressed as

$$\mathbb{P}(B_{\frac{N}{M}}) = \mathbb{P}(I) \cdot \mathbb{P}(I) = (1-p)^{2C_I}.$$

In a GOP, a B-group consists of $(M-1)$ successive B-frames. Thus, we multiply the sum of the probability of each B-group in a GOP with $(M-1)$ in order to calculate the total number of expected successfully decodable B-frames in a GOP. Therefore, the expected number of successfully decodable B-frames of N_{GOP} GOPs can be expressed as

$$\begin{aligned} N_{dec-B} &= (M-1) \cdot \sum_{j=1}^{\frac{N}{M}} \mathbb{P}(B_j) \cdot N_{GOP} \\ &= \left[(\frac{N}{M} - 1) + (1-p)^{C_I} \right] \cdot (1-p)^{C_I} \cdot (M-1) \cdot N_{GOP}. \end{aligned}$$

e) The Expected Decodable Frame Ratio Q: Based on the above model of successfully decodable frames for each frame type within a MPEG video file, Q can be obtained as a function of p:

$$\begin{aligned} Q &= \frac{N_{dec-I} + N_{dec-P} + N_{dec-B}}{N_{total-I} + N_{total-P} + N_{total-B}} \\ &= \frac{\left\{ \frac{N}{M} + \left[(\frac{N}{M} - 1) + (1-p)^{C_I} \right] \cdot (M-1) \right\} \cdot (1-p)^{C_I}}{N}. \end{aligned}$$

2) Ratio of Unencrypted Packets: In a GOP of a MPEG video file, the number of unencrypted packets of I-frame can be expressed as $(1-p) \cdot C_I$. Therefore, we can describe C as follows

$$C = \frac{(1-p) \cdot C_I + C_P \cdot N_P + C_B \cdot N_B}{C_I + C_P \cdot N_P + C_B \cdot N_B}.$$

Clearly, C is a function of the encoded packet ratio p. We can also determine C as the ratio of unencrypted packets in the whole MPEG sequence ignoring small fraction of packets such as MPEG headers.

3) Key Space and Content Expansion: The key space should be large enough to be assigned to the maximum possible number of subscribers. As mentioned in the previous section, we have ${}_S\Pi_s = S^s$ key space. Thus, by choosing S and s carefully, we can construct a sufficiently large key space. If the number of users u in the network is large, the encrypted packets will be increased (its number is $p \cdot C_I$). Therefore, from Eq. in IV-C2, the mean expansion ratio E, which is the average number of encrypted packets for the same video segment of an I-frame, can be obtained by $E = C + (1-C) \cdot f(s, u, S)$, where $f(s, u, S)$ is how many encrypted versions exist for each encrypted packet, which is defined as a function of the number of multiple symmetric keys distributed to a subscriber s, the number of subscribers u, and the number of multiple symmetric keys in total S (managed by the publisher). In order to maximize the cache efficiency, the number of keys which are distributed to subscribers need to be minimal, and thus, we can minimize $f(s, u, S)$ as follows

$$f(s, u, S) = \begin{cases} \dfrac{u+s-1}{s}, & u \le S, \\ \dfrac{S \cdot \alpha + \lceil \frac{u}{S} \rceil + (s - \alpha - 1)}{s}, & u > S. \end{cases}$$

where $\alpha = \lfloor \log_S u \rfloor$ and $u \le S^s$. Clearly, E is a function of the encrypted I-frame packet ratio p and the function $f(s, u, S)$. That is, $E = E(p, s, u, S)$. If $u = S^s$ users are in the network (which is the worst case), the number of differently encrypted versions of an encrypted I-frame packet is $f(s, u, S) = S$.

4) Cache Hit Ratio on a CCN Router: In this subsection, we calculate the cache hit ratio in a CCN router. In [3], the authors assume the request arrival process as a Markov Modulated Rate Process (MMRP) to analyze the cache hit probability at a content object level (They call a content object 'chunk'). This guarantees the request arrival as a Poisson process in terms of content objects and a deterministic process in terms of chunks. In our context, we are interested in the request arrival in a single router at a packet level. Hence, it is reasonable that the packet request arrival follows as a Poisson process in case of multi-content and multi-client at a single router. Therefore, we consider the request arrival process as a Poisson process with mean rate λ. Let K be the total number of packets and let m be the memory size (*i.e.* how many packets can be cached in a router). We assume K different packets are requested with probability $q_k,\ k = 1, ..., K$ and the popularity distribution follows a Zipf distribution [18], i.e. $q_k = c/k^\alpha$ where c is a normalization factor. Hence, the request of packet k is generated according to a Poisson process with rate $\lambda_k = \lambda q_k$.

In order to obtain the cache hit probability, we use the LRU caching policy. Under the independent reference model (IRM), the LRU can be described by an ergodic Markov chain. Let Ω be the state space of this Markov chain. Then, $\Omega \equiv \{1, 2, ..., m\} \cup \{m+1\}$. The left set $\{1, 2, ..., m\}$ means the state space that a given packet k is in the cache with size m and the right set $\{m+1\}$ means the state that the packet is not in the cache. Hence, we obtain the stationary state probabilities of finding packet k in the cache, which is denoted by π^i $(i = 1, 2, ..., m+1)$. Each state corresponds to a particular ordering of m distinct packets within the cache. For the given packet k, we obtain the closed-form model for state $m+1$ by solving

the Markov chain, which represents the cache miss probability for the packet k. Then, we obtain the cache hit probability as

$$P_k^{hit}(m,E) = 1 - \pi_k^{m+1} = 1 - \frac{K^{'} - m}{K^{'}(q_k+1)} \prod_{i=1}^{m-1} \left(\frac{K^{'} - i}{K^{'}(q_k+1) - 1 - i} \right),$$

where $K^{'} = K \cdot E$ ($K^{'} \geq K$) is the total number of packets including the encrypted packets. From this result, we obtain the cache hit probability of the whole $K^{'}$ packets such as

$$P^{hit}(m,E) = \sum_{i=1}^{K^{'}} q_i \cdot P_i^{hit}(m,E).$$

As shown in (IV-C4), the cache hit probability is expressed as a function of two parameters; the storage size m and the expansion ratio E. From the relation $E = E(p,s,u,S)$, the cache hit probability can be determined by the parameters p, s, u and S.

5) Optimization: In this subsection, we formulate the tradeoff model between P^{hit} and Q in terms of m, p, s, u and S. To maximize the cache effectiveness, p should be minimized. However, for data protection, the video segments should be sufficiently encrypted to prevent an adversary from playing the video by increasing p. Therefore, it is necessary to construct a model which can express this tradeoff. Let $T = T(m,p,s,u,S)$ be the tradeoff function, which can be expressed by

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

where γ and δ are scaling parameters. Then, the optimization problem is formulated as follows: for a given decodable frame ratio threshold $\epsilon > 0$, we obtain the maximum cache hit probability by varying the control parameter p in the feasible region such as

$$\begin{aligned} \max_{p} \quad & T \\ s.t. \quad & 0 \leq Q \leq \epsilon \\ & 0 \leq p \leq 1 \\ & 1 \leq u \leq S^s \\ & Q,\ p \in R,\ u \in Z_+. \end{aligned}$$

where R is a set of real numbers and Z_+ is a set of positive integers. With some target minimum value of the video quality ϵ, we can finally obtain desired values of p, s, and S that maximizes the cache hit ratio.

V. Numerical Results

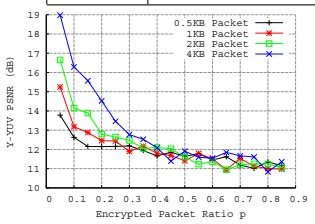
A. Impact of Partial Encryption on PSNR

1) Evaluation Environment: We use `ffmpeg` along with `libavcodec` for encoding and decoding the sample video files, which are "Foreman" and "Akiyo" clips[3], which has 300 frames and its video resolution is 352x288 (CIF). MATLAB is used for PSNR calculation.

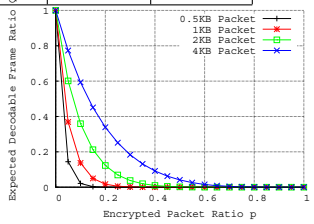
[3]The clips are obtained from the video trace library, http://trace.eas.asu.edu/yuv/index.html.

TABLE II. Statistics of Video Clips

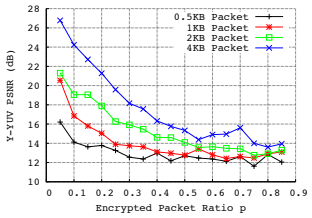
Video File		Foreman	Akiyo
Total number of frames		300	300
I-frames	Number of frames	25	25
	Total size of frames (Bytes)	435,643	312,528
P-frames	Number of frames	75	75
	Total size of frames (Bytes)	245,874	45,859
B-frames	Number of frames	200	200
	Total size of frames (Bytes)	167,196	24,038
C_I	for 0.5K Packet	34.85144	25.00224
	for 1K Packet	17.42572	12.50112
	for 2K Packet	8.71286	6.25056
	for 4K Packet	4.35643	3.12528



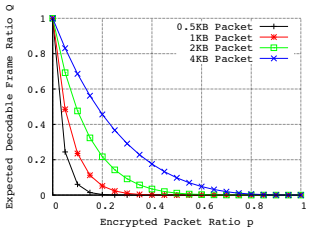
(a) Measured PSNR (Y-YUV) of Foreman CIF, MPEG-4 H.264/AVC, GOP(12, 3)



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



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



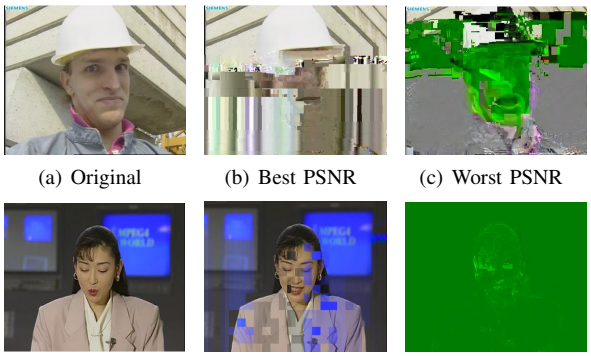
(d) Expected Decodable Frame Ratio Q, Akiyo CIF, GOP(12, 3)

Fig. 4. Comparison of measured PSNR and expected decodable frame ratio Q of the Foreman and Akiyo clip, GOP(12, 3)

2) Evaluation Procedure: At first, we intentionally corrupt the sample file as if it is partially encrypted with an unknown key. Next, we extract I-frames from the sample file, then we split each I-frame into multiple segments of equal length as if they are transmitted as packets. Then we remove the original data of each segment (selected by the ratio p) by overwriting '0x41' on it. After making the pseudo encrypted file, we try to decode it from a standpoint of an adversary. We measure the PSNR value and see if the picture is recognizable to see how much the file is corrupted and whether it is possible to play the file. We compare the YUV format of the two files, one from the original file encoded by the H.264 codec, and the other from the pseudo encrypted file, to calculate the PSNR value.

We repeat this evaluation procedure with different p. Changing p, we calculate the decodable frame rate from the model in Section IV-C1, measure PSNR to identify correlation between these parameters, and find the optimal configuration of the partial encryption. We average the results of 5 runs with different random seeds.

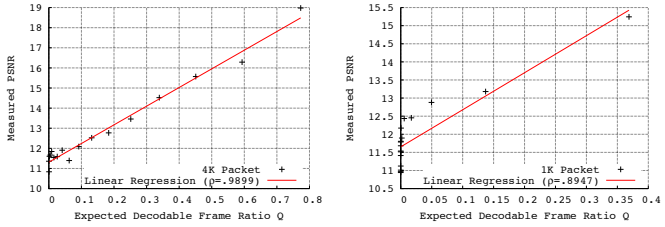
3) Evaluation Results: The video files used in this evaluation are encoded using MPEG-4 H.264 codec with the GOP sequence of `IBBPBBPBBPBB`, which is GOP(12, 3). The statistics of the encoded video files is shown in Table II. We vary p from 0.05 to 1.0, and the packet size from 0.5 KBytes to 4 KBytes. However, the pseudo encrypted files are



(a) Original (b) Best PSNR (c) Worst PSNR

(d) Original (e) Best PSNR (f) Worst PSNR

Fig. 5. Comparison of decoded Foreman ((a)-(c), at frame #100) and Akiyo ((d)-(f), at frame #111) video image captures of Original, Best PSNR, and Worst PSNR case.



(a) 4K Packet (b) 1K Packet

Fig. 6. Measured PSNR vs. expected decodable frame ratio Q (Foreman clip, GOP(12, 3))

hardly playable when $p \geq 0.9$, and hence we plot the results in the range of $0.05 \leq p < 0.9$. In this evaluation, most of the pseudo encrypted files cannot be decoded to the end of frames, and thus the PSNR is measured with decoded frames only. Figures 4(a) and 4(c) show the evaluation results, and Figures 5 compare the original frame, the frame in the best PSNR case (PSNR = 21.073490 for Foreman, and 30.63056 for Akiyo), and the frame in the worst PSNR case (PSNR = 9.001015 for Foreman, and 8.061413 for Akiyo). The captured frames in Figures 5(b), 5(c), 5(e) and 5(f) are not decodable frames in our model. However, the frames are decoded by reconstruction at the decoder. It is difficult to figure out which range of parameter values, *e.g.* p or packet size, would allow the video playable or not. Thus, we verify that the proposed scheme makes the video unplayable effectively to an adversary without appropriate keys.

For comparison purposes, the expected decodable frame rate Q from the model in Section IV-C1 is presented in Figures 4(b) and 4(d) by using the statistics in Table II. From both the modelling and simulation results, we can find that Q and PSNR each tends to decrease as p increases. Each graph in Figure 6 shows that the measured PSNR and Q are in a positive correlation with a high coefficient. Therefore the model can be used to make decision of appropriate p and packet size instead of PSNR value.

B. Impact of Partial Encryption on Cache Hit Probability

1) Evaluation Environment: If we set the packet size to be 4 KBytes and a video file to be 25 GBytes (which is as large as the capacity of a single-layer Blu-Ray Disc), the number of packets of the file is 6.25 million. Also, considering the route processor of Cisco ASR1000 series has up to 4 GBytes

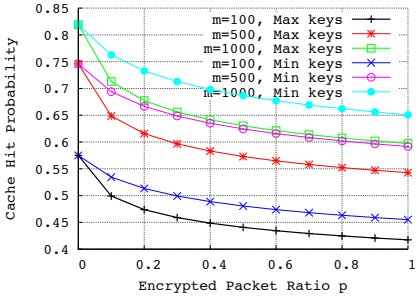


Fig. 7. Comparison of numerical results of a single, independent cache when all the encrypted packets are not shared at all (*Max keys*) and when the encrypted packets are shared as much as possible (*Min keys*) ($S = u = 100$, $s = 3$, I-frame ratio = 0.3)

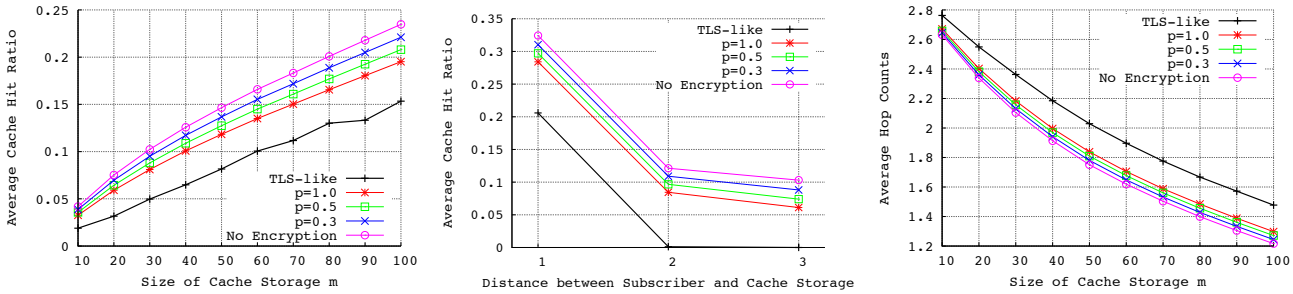
of DRAM, the number of packets for a cache storage (in memory) can contain up to 1 million packets. Keeping the ratio of the content size and the cache storage size consistent, we can simply set the reference content size as 6,250 packets and the reference cache size m as 1,000 packets for the analysis of the model in Section IV-C4. The cache size varies from 100 packets to 1,000 packets.

We also use a packet-level event-driven simulator written in Python to evaluate the cache hit probability. Each subscriber generates the Interests for an arbitrary file as a Poisson process; the inter-arrival times of Interests follow the exponential distribution with $\lambda = 1$ (1/sec). The popularity of the video files is determined by the Zipf distribution [18], whose exponent is set to 1.0. For simulation, we assume that there are 50 scenes and every scene consists of 10 packets with each packet size being 4 KBytes.

A balanced binary tree topology of 15 nodes is used for evaluation. There is only one publisher placed at the root node. Subscribers are placed at the leaf nodes; there are 8 subscribers.

The total number of distinct packets in the network, K, is set to 500 and the I-frame ratio ($= C_I/(C_I + C_P \cdot N_P + C_B \cdot N_B)$) is set to 0.3. We consider two key distribution policies: (i) users do not have common keys among their key sequences, and thus there is no sharing among encrypted I-frame packets, and (ii) users share as many keys as possible to maximize the cache hit ratio. Hence the former and latter cases are expected to exhibit the worst case and the best case scenarios in terms of CCN caching performance. To keep the ratio of the number of packets to the cache size as similar to the numerical analysis, we scale down the reference cache storage size m as 100 packets for simulation. The cache size varies from 10 packets to 100 packets. We run the simulation 5 times with different random seeds, which are averaged.

2) Evaluation Results: Figure 7 shows the numerical results of cache hit probability model at a single, independent cache storage. The expected cache hit probability decreases as p increases and m decreases. Depending on key distribution policies, the expected cache hit probability is also affected. By choosing an optimal key sharing policy, the higher cache hit probability is achieved when other parameters stay the same. Therefore there is a tradeoff between the cache hit probability and p. Note that the cache hit probability also depends on the key distribution policies.



(a) Average cache hit ratio

(b) Average cache hit ratio by the distance to the cache storage (m = 100)

(c) Average hop counts

Fig. 8. Comparison of cache hit ratio and hop counts from simulation results (K = 500, I-frame ratio = 0.3)

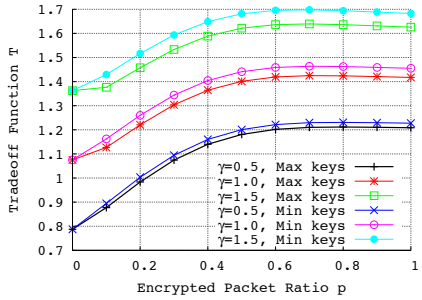


Fig. 9. Comparison of numerical results of the tradeoff function T when all the encrypted packets are not shared at all (*Max keys*) and when the encrypted packets are shared as much as possible (*Min keys*) (δ = 1.0, S = u = 100, s = 3, I-frame ratio = 0.3, K = 6250, m = 100, GOP(12, 3), C_I = 4.35643)

In Figure 8, we present the simulation results of the cache hit ratio and hop count. Average cache hit ratio[4] and hop count of the proposed partial encryption is comparable to 'No Encyption'. Even when p is set to 1.0, the proposed partial encryption performs much better than 'TLS-like' encryption scheme, which means per-subscriber full encryption scheme.

The cache hit ratio increases as the cache size increases, and as the distance between the subscribers and the cache storage becomes closer. Figure 8(b) shows that the closest cache storage of each subscriber exhibits the highest cache hit ratio. It also validates our model of the cache hit probability for a single, independent cache described in Section IV-C4 since it can explain the network behaviors related to the cache hit probability and encryption. The average hop count decreases as the cache size increases, which means that the average hop count and the average cache hit ratio are in negative correlation.

C. Finding Optimal Configurations Considering the Tradeoff

We seek to obtain the optimal values of the parameters as modelled in Section IV-C5. While changing the scaling parameter γ from 0.5 to 1.5, we also vary the encrypted packet ratio p from 0.0 to 1.0 and two different key distribution policies are applied.

The numerical results are shown in Figure 9. With the configuration of δ, S, s, u, I-frame ratio, K, m, N, M, and C_I of Figure 9, we can see the tradeoff function T is maximized near p = 0.7. We can see that the optimal value of p is slightly decreased as γ is increased because γ is a weight of the cache hit probability. Considering the feasible region as described in Section IV-C5, the optimal point can be changed as the conditions on the region change.

[4]Note that the cache hit ratio is measured after the warm-up period, which means the measurement begins right after the cache becomes full for the first time.

VI. Related Work

We introduce relevant studies that deal with access control mechanisms for content-oriented networking and video encryption.

A. Access Control for Content-oriented Networking

For content-oriented networking, the Access Control Enforcement Delegation scheme [8] is proposed for the PURSUIT project of EU FP7. Its main idea comes from OAuth which is broadly used on the current Internet. It depends on a security feature of the PURSUIT naming system, so it is not applicable to CCN.

For encrypted video contents, broadcast encryption [7] is designed to deliver encrypted video contents, such as TV programs, via a broadcast channel. The subscribers receive the same encrypted contents, and the same key is distributed for legitimate subscribers. It is more suitable for real-time streaming rather than VoD services and it is hard to detect key leakage.

Group signature [5] is a good fit for VoD services compared to broadcast encryption. This scheme is intended to provide a certain level of anonymity via a group key. If legal subscribers belong to in a group, we can use the group signature for content distribution via CCN. Since the members share the same group key, in-network caching for CCN will be effective. However, the heavy dependence on the group manager may hurdle the actual deployment.

Proxy re-encryption [2] is good for reuse of encrypted data in caches. With the transformation key, the authorized user A can change the encrypted content which is encrypted for user B, so that user A can decrypt the content with her private key. The proxy re-encryption is also plausible for content distribution in CCN [1, 4]. However, it requires many public key calculations, which consumes substantial computational power.

B. Video Encryption Schemes

1) Selective Encryption: Selective encryption concept is used for our proposal. It leverages different priorities of MPEG frames.

Secure MPEG (SECMPEG) [15] provides four different levels of security (*i.e.* data protection), from the headers-only encryption to the whole MPEG sequence encryption, to reduce the complexity of computations. This is the first technique to realize the benefits of selective encryption but a special encoder/decoder is required [12].

Aegis [17] encrypts all the I-frames and the sequence headers, while B- and P-frames are left unencrypted. This scheme provides sufficient confidentiality to protect the MPEG video from unauthorized access. Our proposal is influenced by Aegis in that B- and P-frames are unencrypted.

2) Data Corruption: [9, 10] propose a MPEG video corruption mechanism in order to use caching and pre-distribution. The correct parts of video can be distributed by unicasting to the authorized user and the intentionally corrupted parts can be replaced with the correct ones. [10] finds that with a destruction ratio of 1%, the video is unplayable due to the error propagation. This results also support effectiveness of our partial encryption scheme.

VII. Conclusion

Video traffic in the Internet is expected to overwhelm in the near future. Content-centric networking (CCN) is a clean slate network architecture, which may be able to handle the video traffic by efficient content delivery. Assuming MPEG video streams, we seek to achieve data protection while preserving the advantage of CCN's in-network caching. We present a CCN security framework for video streaming services, whose key mechanism is the partial encryption that trades off between the data protection and caching efficiency in CCN. The proposed framework also deals with privacy, access control, and user authentication. Based on the framework, the mathematical model is developed to find the optimal configurations of parameters. The validity of the model is also confirmed by simulation, which analyzes the relation between video quality, key management, and cache effectiveness. Also the proposed framework can be extended to tracing who is responsible for the key leakage if all the I-frame packets are encrypted and watermarked, which is left for future work.

References

[1] C. Abdelberi, E. D. Cristofaro, M. A. Kâafar, and E. Uzun, "Privacy in content-oriented networking: Threats and countermeasures," *CoRR*, vol. abs/1211.5183, 2012.

[2] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology — EUROCRYPT'98*, ser. Lecture Notes in Computer Science, K. Nyberg, Ed. Springer Berlin Heidelberg, 1998, vol. 1403, pp. 127–144. [Online]. Available: http://dx.doi.org/10.1007/BFb0054122

[3] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proc. of ITC*, ser. ITC '11. ITCP, 2011, pp. 111–118. [Online]. Available: http://dl.acm.org/citation.cfm?id=2043468.2043487

[4] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in content-oriented networking: threats and countermeasures," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 25–33, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2500098.2500102

[5] D. Chaum and E. Heyst, "Group signatures," in *Advances in Cryptology — EUROCRYPT'91*, ser. Lecture Notes in Computer Science, D. Davies, Ed. Springer Berlin Heidelberg, 1991, vol. 547, pp. 257–265. [Online]. Available: http://dx.doi.org/10.1007/3-540-46416-6_22

[6] Cisco, "The zettabyte era–trends and analysis," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.html.

[7] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology — CRYPTO'93*, ser. Lecture Notes in Computer Science, D. Stinson, Ed. Springer Berlin Heidelberg, 1994, vol. 773, pp. 480–491. [Online]. Available: http://dx.doi.org/10.1007/3-540-48329-2_40

[8] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," in *Proc. of ICN 2012*, ser. ICN '12. New York, NY, USA: ACM, 2012, pp. 85–90. [Online]. Available: http://doi.acm.org/10.1145/2342488.2342507

[9] C. Griwodz, "Video protection by partial content corruption," in *Multimedia and Security Workshop at ACM Multimedia*, pp. 37–39.

[10] C. Griwodz, O. Merkel, J. Dittmann, and R. Steinmetz, "Protecting vod the easier way," in *Proc. of the sixth ACM MM*. 290751: ACM, pp. 21–28.

[11] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2063176.2063204

[12] K. Kalaivani and B. Sivakumar, "Survey on multimedia data security," *International Journal of Modeling and Optimization*, vol. 2, no. 1, pp. 36–41, 2012.

[13] H. Koumaras, C. H. Lin, C. K. Shieh, and A. Kourtis, "A framework for end-to-end video quality prediction of mpeg video," *Journal of Visual Communication and Image Representation*, vol. 21, no. 2, pp. 139–154, 2010.

[14] D. Le Gall, "Mpeg: a video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, Apr. 1991. [Online]. Available: http://doi.acm.org/10.1145/103085.103090

[15] J. Meyer and F. Gadegast, "Securitymechanisms for multimedia-data with the example mpeg-i-video," in *Projectdescription SECMPEG*. Technical University of Berlin, 1995.

[16] H. Schulze and K. Mochalski, "ipoque - internet study 2008/2009," http://www.ipoque.com/en/resources/internet-studies.

[17] G. Spanos and T. Maples, "Performance study of a selective encryption scheme for the security of networked, real-time video," in *Proc. of ICCCN*, 1995, pp. 2–10.

[18] G. K. Zipf, *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press, 1932.