International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

Novel Video Content Encryption approach for In Network Services

Puviyarasu R, Meenambika A, Gopalakrishnan H, Sruthi M, Dineshkumar U

Department of Computer Science and Engineering, Akshaya College of Engineering and Technology, Coimbatore, Tamil Nadu, India

ABSTRACT

The modern view of multimedia communications like the video transmission and video streaming allows the people to access the stored media using the network around the world. Video streaming is a type of multimedia that is constantly received and presented to an end-user while being delivered by a video service provider or sender. This existing system is a secure Redundancy Elimination protocol to leverage encrypted in-network caching for video delivery without knowing the underlying video content and fingerprint. In case of large media size, then the video encryption requires large video handling time and video delivery time such as transmission latency and packet transmission Jitter. In a novel video content encryption approach based on video frames, which encrypts the individual frame and attaches the corresponding macro block information by eliminating the redundant frame. It handles the videos frames using H-264 encoding model, the actual videos frame content is divided into I, P and B-frames providing the video information in terms of Intra Coded Pictures, Predicted pictures and Bi-directionally predicted pictures. The process is repeated by processing the complete file into number of video frames. The proposed frame based encryption system achieves the better performances compare to chunk based encryption in terms of Transmission latency, Communication reliability and Packet Jitter.

Keywords : Novel Video Content Encryption, Network Services, Video Service Provider, H-264, Packet Jitter, Transmission Latency, Communication Reliability, Encryption

I. INTRODUCTION

Streaming media is multimedia that is constantly received by and presented to an end-user while being delivered by a provider. A client end-user can use their media player to begin to play the data file before the entire file has been transmitted. Distinguishing delivery method from the media distributed applies specifically to telecommunications networks, as most of the delivery systems are either inherently streaming or inherently non-streaming. Multicast protocols were developed to reduce the server/network loads resulting from duplicate data streams that occur when many recipients receive unicast content streams independently. These protocols send a single stream from the source to a group of recipients. Depending on the network infrastructure and type, multicast transmission may or may not be feasible Telecoms network constitutes a accumulation of terminal clients, connections are associated so as to enable telecom between the terminals. The transmission system links associate the clients together. The nodes use circuit exchanging, message switching or packet switching to pass the signal through the correct links and clients to accomplish the correct destination terminal.

Internet network: access to the network allows users to use many resources. Over time the Internet network will replace books. This will enable users to discover information almost instantly and apply concepts to different situations. The Internet can be used for recreational, governmental, educational, and other purposes. Businesses in particular use the Internet network for research or to service customers and clients. The data network is used extensively throughout the world to connect individuals and organizations. Data networks can be connected to allow users seamless access to resources that are hosted

outside of the particular provider they are connected to. The Internet is the best example of many data networks from different organizations all operating under a single address space.

In such networks, users focus only on contents, rather than the physical locations from which contents can be retrieved. Moreover, the network routing and in-network storage are most likely provisioned by the same network carrier in a content-centric manner. Hence, the cost of coordinating content caching and the performance of routing traffic become main concerns in CCN. On the other hand, non-coordinated caching mechanisms store only the locally most popular contents at each CCN router, without coordination with other routers. Therefore, such mechanisms not only incur less coordination cost but also are more likely to store less distinct contents due to lack of coordination. Furthermore, studies have shown that the popularity of both Web and video contents follows the Zipf distribution.

Therefore, there are clear trade-offs between the network performance and the coordination cost when designing in-network caching mechanisms for CCN. More specifically, coordinated caching mechanisms may trade the coordination cost for the network performance, while non-coordinated caching mechanisms may incur a significantly lower cost on provisioning in-network caching and may degrade the network performance due to lack of fine-grained control on where contents are cached, retrieved and routed to users.

Despite being promising, in-network content caching would also raise new security and privacy concerns, due to the increasingly untrusted networked environments, where the network caching devices are not necessarily in the same trusted domain as the content users or under the full control of content providers. For users, it is crucial to ensure their private data, like video access history, subscription details, or even personal videos, not to be exposed unwillingly. For content providers, strong protection against unauthorized content access or copyright infringement is a commercial imperative.

In the proposed system, the traditional view of video transmission enabled networks often looks for tangible value in terms of cost savings, especially reduced traffic expenses. There are many techniques for encrypting the video data like selective encryption in different steps of compression video. Besides, the system also supports encrypted dynamic cache management and one to many access control protocols. Delay in video handling occurs because of the chunk based encryption process without knowing the underlying video frame content. This frame information contains the redundant video content as Macro blocks. This signature code generation process is repeated by processing the complete file into number of video frames.

II. METHODS AND MATERIAL

1. Related work

In-network caching necessitates the transformation of centralized operations of traditional, overlay caching techniques to a decentralized and uncoordinated environment. Given that caching capacity in routers is relatively small in comparison to the amount of forwarded content, a key aspect is balanced distribution of content among the available caches. In this paper, we are concerned with decentralized, real-time distribution of content in router caches. Our goal is to reduce caching redundancy and in turn, make more efficient utilization of available cache resources along a delivery path. The in-network caching scheme, called ProbCache, approximates the caching capability of a path and caches contents probabilistically in order to: i) leave caching space for other flows sharing (part of) the same path, and ii) fairly multiplex contents of different flows among caches of a shared path.

Although the volume of Web traffic on the Internet is staggering, a large percentage of that traffic is redundant-multiple users at any given site request much of the same content. This means that a significant percentage of the WAN infrastructure carries the identical content (and identical requests for it) day after day. Eliminating a significant amount of recurring telecommunications charges offers an enormous savings opportunity for enterprise and service provider customers. Web caching performs the local storage of Web content to serve these redundant user requests more quickly, without sending the requests and the resulting content over the WAN.

Browser Freshness Controls

Finally, clients can always explicitly refresh content at any time by using the browser's reload/refresh button. The reload/refresh command is a browser-triggered command to request a data refresh. A reload/refresh will issue a series of IMS requests asking for only data that has changed. The shift+reload/shift+refresh command is an extension of the reload/refresh command. In correctly implemented browsers, this command always triggers a "pragma: no cache" rather than an IMS request. As a result, cache engines are bypassed and the end server directly fulfills all content.

Reverse Proxy Caching

Cache engines are frequently deployed nearby clients to ensure faster network response time and minimal WAN bandwidth usage. Thus, the caches are caching the clients' most frequently accessed content. In addition, cache engines can also be deployed in front of Web server farms to increase the server farm capacity and improve Web site performance. This configuration is called reverse proxy caching because the cache engines are only caching content from the servers for whom they are acting as a front-end.

This feature is particularly important when cache engines are acting as a front-end for server farms in which certain content is dramatically more popular than other content on the servers. Using reverse-proxy caching allows administrators to prevent a small number high-demand URLs from impacting overall server performance. Better yet, this means the high-demand URLs do not have to be identified, manually replicated, or independently managed from the bulk of the URLs on the servers.

WCCP Network Caching

- Cisco developed WCCP, a router-cache protocol that localizes network traffic and provides "network-intelligent" load distribution across multiple network caches for maximized download performance and content availability.
- The cache component of the Cisco caching solution comprises network-integrated caching solutions-the Cisco Cache Engine 500 Series. They are network-integrated because they:
- Provide network management capabilities already available on traditional Cisco networking gear (such as Cisco IOS CLI and RADIUS support), resulting in minimized management and operational costs.

Are inherently designed and implemented as caching-specific networking hardware, rather than being standalone server platforms adapted as caches. Thus, the high-density Cisco Cache Engines physically integrate better into the network infrastructure as network extensions transparently insert into existing network infrastructures and adapt to unusual network conditions, resulting in minimized deployment and operational costs and greater content availability.

Network-Based Shared Caching

The cache engine was designed from the ground up as a loosely coupled, multinode network system optimized to provide robust shared network caching. The cache engine solution comprises the Web Cache Control Protocol and one or more Cisco cache engines that store the data in the local network. The Web Cache Control Protocol defines the communication between the cache engine and the router. Using the Web Cache Control Protocol, the router directs only Web requests to the cache engine. The router also determines cache engine availability, and redirects requests to new cache engines as they are added to an installation.

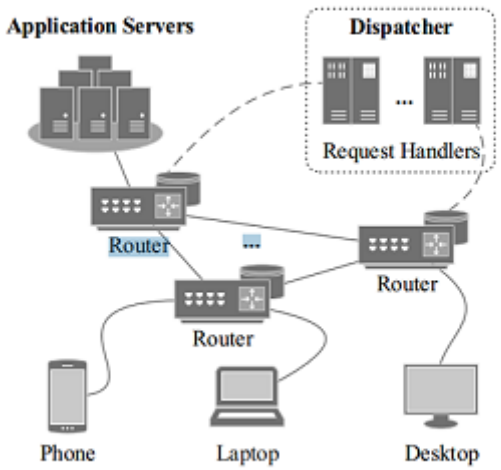


Figure 1. System Architecture

Dynamic Client Bypass

Some Web sites require clients to be authenticated using the client's IP address. However, when a network

cache is inserted between a client and a Web server, the Web server only sees the cache's IP address and not the client's IP address. To overcome this issue and similar situations, the Cisco Cache Engine has a dynamic client bypass feature that effectively allows clients, under certain conditions, to bypass cache engines and directly connect to origin Web servers. The result is that a Cisco Cache Engine can preserve existing source IP authentication models and pass through server error messages to clients. Because the cache engine dynamically adapts to these situations, less management is required to ensure cache transparency.

2. In-Network Caching

The networked system for efficient encrypted video delivery is designed while preserving the benefits of in-network caching. As video chunks are encrypted before distribution and a secure redundancy elimination protocol to enable fast video delivery via leveraging cached encrypted chunks. In order to scale with the increasing amount of video traffic, modern network architectures leverage a core technique called in-network caching. Content is stored in the cache-enabled network devices, and then forwarded based on users' requests.

The first threat is introduced by the new functionality at routers in future network architectures. For instances, the cache-enabled routers with computing and storage abilities may expose a much large attack surface than traditional routers. Once they are hacked, the cached content is directly exposed. The adversary can not only eavesdrop on users' request histories, but also commit video piracy. The second threat comes from the network infrastructure. It is usually operated by Internet service providers (ISPs) or other third-party CDN providers. While dutifully delivering the network services, those service providers or the network administrators inside may also be interested in the video content and the user privacy. Therefore, they are not necessarily in the same trusted domain as the users or the video content providers. Traditional copyright protection like digital watermarking can still be employed to trace the users against privacy. Currently, we emphasize on the protection of video confidentiality and the counter measure of unauthorized access, and thus the topics on regular network threats like DDoS, video integrity, and content flow verification. The system is designed to leverage the encrypted in-network caching for video delivery without compromising the confidentiality of the video content or other sensitive information, e.g., video fingerprints, video names, and so on. To ensure strong protection, the video content or chunks remain in encrypted forms during their lifetime in the network. As a result, even the network devices are hacked, the damage only confines to the cipher text, i.e., neither extending to users' sensitive data nor the video content.

It should achieve the security strength in such a way that: 1) Without a controlled capability, i.e., a secure and valid fingerprint token, the adversary cannot get any useful information about video content and fingerprints. 2) Processing a secure token over the encrypted index reveals at most the chunk pseudonyms of the requested video, and no other information. In particular, the security strength in our system should be precisely captured, the formal security analysis should be presented.

From a high level point of view, our networked system includes the following procedures to provide secure and efficient video delivery via encrypted in-network caching. Initially, the application server builds an encrypted fingerprint index for the request handler, which stores encrypted pseudonyms of video chunks. When the application server receives a user request, it will generate a secure token from the fingerprint of the requested video and send it to the request handler. Upon receiving the token, the request handler processes it over the encrypted index, and obtains the pseudonyms of encrypted chunks. In addition, the request handler maintains the in-network distribution of encrypted chunks for fast routing. If those chunks are not cached in the network yet, the application server will directly distribute them via the network to the user.

(b) Video delivery via encrypted in-network caching
Fig. 2: The proposed secure RE protocol

For security guarantees, the system choose to perform each item insertion in a random fashion by using pseudo-random function (PRF); that is, the video fingerprint and the sequence of the chunk are used as the inputs of PRFs to seek available buckets for each item. Moreover, we must encrypt all the buckets so that the request handler cannot have any useful information from the encrypted index. To compact the index size, our designed index only stores the chunk pseudonyms. By encoding the fingerprint into a random mask to encrypt the data, we do not have to separately store the

fingerprint and save index space. Those chunks will be simultaneously cached at routers through any known cache placement strategies probabilistic caching. Otherwise, the request handler will locate the targeted chunks, and facilitate the routers to send them to the user.

Drawbacks

- If the file size is large then signature code size is high.
- It will increase the complexity of the encryption process.
- The method does not provide validation process to secure the data from intruders, hijackers and overhearing devices.
- The connectivity model took long time to complete the authentication and crypt operations.

3. Frame based Encryption

The proposed system is used to protect different multimedia content types including videos, images and audio. It is deployed in both public and private in-network. It creates signatures for multimedia content, and distributed matching engine for multimedia objects. The signature is generated based on the depth signal in the video like spectrum value of the audio signal. Multi-level signature generation process is used to design the efficient encryption for multimedia content. This code generation process is repeated by processing the complete file into number of chunks Individual signature codes are merged by using logical operation such as XOR operation. The response of a local Web cache is often three times faster than the download time for the same content over the WAN. End users see dramatic improvements in response times, and the implementation is completely transparent to them.

A Web cache stores Web pages and content on a storage device that is physically or logically closer to the user-closer and faster than a Web lookup. By reducing the amount of traffic on WAN links and on overburdened Web servers, caching provides significant benefits to ISPs, enterprise networks, and end users. The first step in creating a network-integrated cache engine is to ensure that the network supports traffic localization, which can be achieved by enabling content routing technology at the system-level, and setting specific parameters to optimize network traffic. Once the right network foundation is in place, network caches are added into strategic points within the existing network. By pairing software and hardware, Cisco creates a network-integrated cache engine. Network-integrated caches have at least the following three properties:

- Managed like networking equipment, resulting in minimized operational costs
- Designed like high-density networking hardware, resulting in better physical integration into the network infrastructure as network extensions and minimizing costs associated with leasing rack space
- Transparently inserted into the network, resulting in minimized deployment and operational costs and greater content availability

Proxy servers are software applications that run on general-purpose hardware and operating systems. A proxy server is placed on hardware that is physically between a client application, such as a Web browser, and a Web server. The proxy acts as a gatekeeper that receives all packets destined for the Web server and examines each packet to determine if it can fulfill the requests itself; if not, it makes its own request to the Web server. Proxy servers can also be used to filter requests, for example, to prevent its employees from accessing a specific set of Web sites. Unfortunately, proxy servers are not optimized for caching, and do not scale under heavy network loads. In addition, because the proxy is in the path of all user traffic, two problems arise: all traffic is slowed to allow the proxy to examine each packet, and failure of the proxy software or hardware causes all users to lose network access. Expensive hardware is required to compensate for the low software performance and the lack of scalability of proxy servers.

When the storage in the router is full, our system can also utilize existing strategies for cache eviction. For instances, the system can apply first-in first-out (FIFO), least-recently used, time-to-live (TTL) strategies. When the encrypted chunks are evicted, the routers are asked to update the information to the request handler for consistency. Regarding our proposed RE protocol, when some chunks in the manifest are evicted, the router will return Null back to the user. In that case, the user will be asked to resend the request to the application server. Then the application server will

generate secure fingerprint tokens to the request handler to obtain the latest manifest. Recall that when the chunks are placed or evicted, the routers will send the feedback to the request handler for the update of chunk distribution. After that, if those evicted chunks are found in the network somewhere else, they will be sent back to the user from the targeted routers. Otherwise, they will be downloaded from the application servers.

On the protection of video lengths: We note that the number of chunks and the total sizes for a video will be revealed in the secure RE protocol. If the adversary has some background knowledge about the videos, such information can be exploited to compromise the video confidentiality. For instances, if the duration of each chunk is known, the overall time for a video will be learned. Since each chunk is encrypted via symmetric encryption, the total size of encrypted chunks is roughly equal to the size of the original video. In general, such side-channel information is not content related. But when the video duration or the video length is considered to be sensitive, we can simply add random chunks to obfuscate the video sizes.

4. System Evaluation

Deployment key generation

- Any user is registered in private cloud key is shared between cloud system and user.
- This shared key is generated from user own key and system key.
- This shared key contain domain parameters, group id, encryption key, hash code (MD5 hash) of key for key verification

Performing asymmetric encryption and decryption

- First, each user will be issued private keys for their corresponding privileges. These private keys can be applied by the user to generate file token for duplicate check.
- However, during file uploading, the user needs to compute file tokens for sharing with other users with privileges.
- To compute these file tokens, the user needs to know the private keys for group members which mean file tokes could only be chosen from privileged users.

Tag-generation and Update new session key

- Unauthorized users without appropriate privileges or files, including the S-CSP and the private cloud server, should be prevented from access to the underlying plaintext stored at S-CSP.
- For each communication, new key is updated with key chain process, by combining previous key and new parameter received from cloud system

Update new key during user revoking procedure

- Key timer is maintained for all users. If the key is expired based on time, then the new key is generated and shared with the server.
- And in case of any user revoked from group or any user added to group then new key is generated to maintain security.

Most products that use a "hashing" mechanism also require an index to store the hashes so that they can be looked up quickly to compare against new hashes to see if the new data is unique (i.e., not already stored), or there is a hash match and the new data element does not need to be stored. These indexes must be very fast or handled in such a manner that the unique data stored increases and becomes fragmented so that the solution doesn't slow down during the hash lookup and compare process.

Different solutions from various vendors use diverse hashing algorithms, but the process is basically the same. The term "hashing the data" means "creating a mathematical representation of a specific dataset that can be statistically guaranteed to be unique from any other dataset." The way this is done is to use a generally understood and approved method to encrypt each dataset, so that the metadata or resulting mathematical encryption "hash" can be used to either reproduce the original data or as a lookup within the index to see if any new data hashes compare to any stored data hashes, so the new data can be ignored.

Hashing-base solutions typically provide great results in reducing storage requirements for a particular data set, but there is one huge disadvantage over delta versioning. Since everything is stored as a jumble of mathematical hashes, objects and indexes, it requires the data to be "re-constituted" prior to being usable again for applications. This re-constitution process

takes time, which may have a negative impact if the data needs to be recovered NOW. Micro-scanning solutions have a slightly lower overall ratio for a particular dataset, but the data is always in the native format of the application and is always immediately available for use. This is important when quick application recovery is the goal. Another benefit of micro scanning is the ability to restore only the sectors required to recover any lost or corrupted data, so massive databases like data warehouses can sometimes be recovered over the network almost instantly.

III. RESULTS AND DISCUSSION

SYSTEM IMPLEMENTATION

In-network computing is the long dreamed vision of computing as a utility, where in-network customers can remotely store their data into the in-network so as to enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources. To protect data privacy and combat unsolicited accesses in the in-network and beyond, sensitive data, may have to be encrypted by data owners before outsourcing to the commercial public in-network.

The system must preserve the following objectives,

- The main objective is to decreasing the processing time of the system for encrypting the video files in both small size and large size.
- To reducing the video caching latency and to utilize the storage by leveraging cached encrypted chunks.

The video frame based encryption methodology is applied in the following modules

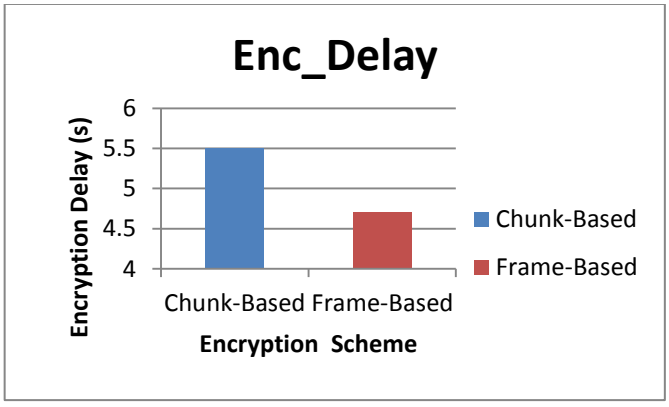
- Providing in-network services and create connectivity with mobile devices
- Accessing in-network server using credential information and store the multimedia content without encryption
- Generating the signature code for video and perform the video encryption by applying the chunk based splitting process
- Dividing the video into frames and to Identifying and eliminating the redundant copies by validating the video frames.

Providing in-network services and creates connectivity with mobile devices

Setting up the network server and to create the web interface and mobile interface for connection maintenance. Web service is created using the internet information service and the transceiver devices are registered with the network server using username and password. Server configuration algorithm is used setting up the in-network server in the existing system. The proposed system is designed by credentials verification algorithm with multi device recognition algorithm. The system takes the Ip address of the server and device along with the Credentials of the user device as Input parameter. The system produces the Connection identifier generated for each user device.

Accessing in-network server using credential information and store the multimedia content without encryption

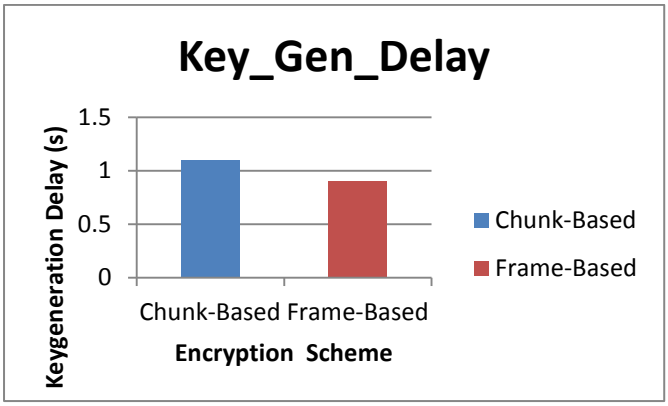
Creating the ftp connection between sender and receivers. Exchanging the file and downloading the video file. The establishment of ftp connection between sender and receivers are registered by verifying the credentials and validation of available service port. Connection id and File name verification algorithm is used to exchange the file between devices. File content Deduplication algorithm is used to apply the file exchange operations. It verifies the signature code of the file before uploading into the ftp server. Credentials, IP address and file content of Sender, receivers devices are considered as the input parameters and the Uploaded file ID and index of the file are generated as output. Successful file exchange operation is applied by checking the file size and storage availability.



	Chunk-Based	Frame-Based
Enc_Delay	5.5	4.7

Generating the Key and chunk code for video and perform the video encryption by applying the chunk based splitting process

The system is designed to create the dedicated key and shared key between the sender and receivers. The uploaded file content is encrypted using the shared public key. And receiver device decrypts the data using private key. Elliptic curve key generation algorithm is used to generate the shared key between users. The file content is encrypted and decrypted using the shared key. Device ID and session information with plain text data of the video content are taken as input parameter and Dedicated key and Shared key is generated and the encrypted video content is transmitted between devices.

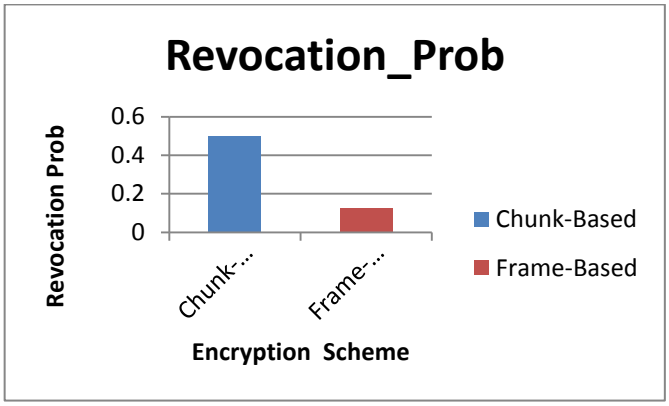


	Chunk-Based	Frame-Based
Key_Gen_Delay	1.1	0.9

Dividing the video into frames and to identify with the elimination of the redundant copies by validating the video frames.

Chunk based encryption is applied to encrypt the video data. The chunks are subdivided into number of frames for the frame processing model. Frame based encryption with the handling of macro blocks of video data is applied by differentiating the frame type content. The system sub divides the macro block structure and to provide the ease of encryption modeling of video data. Video content in terms of chunk and frame content of video data are taken as input and the Encrypted video content with successful decryption are obtained as output and signature generated for validating the decryption of data.

Most products that use a "hashing" mechanism also require an index to store the hashes so that they can be looked up quickly to compare against new hashes to see if the new data is unique (i.e., not already stored), or there is a hash match and the new data element does not need to be stored. These indexes must be very fast or handled in such a manner that the unique data stored increases and becomes fragmented so that the solution doesn't slow down during the hash lookup and compare process.



	Chunk-Based	Frame-Based
Revocation Prob	0.5	0.125

Different solutions from various vendors use diverse hashing algorithms, but the process is basically the same. The term "hashing the data" means "creating a mathematical representation of a specific dataset that can be statistically guaranteed to be unique from any other dataset." The way this is done is to use a generally understood and approved method to encrypt each dataset, so that the metadata or resulting mathematical encryption "hash" can be used to either reproduce the original data or as a lookup within the index to see if any new data hashes compare to any stored data hashes, so the new data can be ignored.

Hashing-based dedupe solutions typically provide great results in reducing storage requirements for a particular data set, but there is one huge disadvantage over delta versioning. Since everything is stored as a jumble of mathematical hashes, objects and indexes, it requires the data to be "re-constituted" prior to being usable again for applications. This re-constitution process takes time, which may have a negative impact if the data needs to be recovered NOW. Micro-scanning

solutions have a slightly lower overall ratio for a particular dataset, but the data is always in the native format of the application and is always immediately available for use. This is important when quick application recovery is the goal. Another benefit of micro scanning is the ability to restore only the sectors required to recover any lost or corrupted data, so massive databases like data warehouses can sometimes be recovered over the network almost instantly.

IV.CONCLUSION

Video streaming is a type of multimedia that is constantly received and presented to an end-user while being delivered by a video service provider or sender. This existing system is a secure Redundancy Elimination protocol to leverage encrypted in-network caching for video. In case of large media size, then the video encryption requires large video handling time and video delivery time such as transmission latency and packet transmission Jitter. In this work, novel video content encryption approach is proposed based on video frames, which encrypts the individual frame and attaches the corresponding macro block information by eliminating the redundant frame. It handles the videos frames using H-264 encoding model, the actual videos frame content is divided into I, P and B-frames providing the video information. The process is repeated by processing the complete file into number of video frames. The proposed frame based encryption system achieved the better performances compare to chunk based encryption in terms of Transmission latency, Communication reliability and Packet Jitter.

V. REFERENCES

[1] Cisco, "Virtual Networking Index (VNI)," http://www.cisco.com/web/solutions/sp/vni/vni forecast highlights index.html, 2015.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman,"A survey of information-centric networking," IEEE Communications Magazine, vol. 50, no. 7, pp. 26–36, 2012.

[3] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router designand implications," in Proc. of the ACM Re-Architecting the Internet Workshop, 2010.

[4] W. So, A. Narayanan, and D. Oran, "Named data networking on arouter: fast and dos-resistant forwarding with hash tables," in Proc. Of ACM IEEE ANCS, 2013.

[5] Cisco, "Cisco ASR 9000 Series Integrated Service Module,"onlineat:http://www.cisco.com/c/en/us/ products/collateral/routers/asr-9000-series-aggregation-services-routers/data sheet c78-663164.pdf, 2012.

[6] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network cachingfor information-centric networks," in Proc. of the second edition of the ACM ICN workshop on Information-centric networking, 2012.

[7] D. Perino, M. Varvello, and K. P. Puttaswamy, "ICN-RE: redundancy elimination for information-centric networking," in Proc. of the second edition of the ICN workshop on Information-centric networking, 2012.

[8] A. Chanda and C. Westphal, "Contentflow: Mapping content to flows in software defined networks," in Proc. of IEEE Globecom, 2013.

[9] Y. Sun, S. K. Fayaz, Y. Guo, V. Sekar, Y. Jin, M. A. Kaafar, and S. Uhlig, "Trace-driven analysis of icn caching algorithms on video-on-demand workloads," in Proc. of ACM CoNEXT, 2014.

[10] N. A. Jagadeesan, R. Pal, K. Nadikuditi, Y. Huang, E. Shi, and M. Yu, "A secure computation framework for sdns," in Proc. of ACM HotSDN,2014.

[11] R. Wang, Y. Shoshitaishvili, C. Kruegel, and G. Vigna, "Steal this movie: Automatically bypassing drm protection in streaming media services."in Proc. of USENIX Security, 2013.

[12] S. Misra, R. Tourani, and N. E. Majd, "Secure content delivery in information-centric networks: design, implementation, and analyses," in Proc. of the 3rd ACM SIGCOMM workshop on Information-centric networking, 2013.

[13] J. Liang, J. Jiang, H. Duan, K. Li, T. Wan, and J. Wu, "When HTTPS meets CDN: A case of authentication in delegated service," in Proc. Of IEEE S&P, 2014.

[14] F. Angius, C. Westphal, M. Gerla, and G. Pau, "Drop dead data: What to expect securing data instead of channels," in Proc. of IEEE CCNC,2015.

[15] D. Dorwin, A. Bateman, and M. Watson, "W3c editor's draft: Encrypted media extensions," on line at: https://w3c.github.io/encrypted-media/, 2015.

[16] J. Sherry, C. Lan, A. R. P. Popa, and S. Ratnasamy, "Blindbox: Deep packet inspection for encrypted traffic," in Proc. of ACM SIGCOMM,2015.

[17] ISO/IEC 23009-1:2014, "Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," 2014.

[18] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," Journal of Computer Security, vol. 19, no. 5, pp. 895–934, 2011.

[19] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in Proc. of ACM CCS, 2012.

[20] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo filter: Practically better than bloom," in Proc. of ACM CoNEXT,2014.

[21] ISO/IEC 23008-1:2014, "Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 1:MPEG media transport (MMT)," 2014.

[22] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM TISSEC, vol. 9, no. 1, pp. 1–30, 2006.

[23] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: network processing as a cloud service," ACM SIGCOMM Computer CommunicationReview, vol. 42, no. 4, pp. 13–24, 2012.

[24] S. Manfredi, F. Oliviero, and S. P. Romano, "A distributed control law for load balancing in content delivery networks," IEEE/ACM TON, vol. 21, no. 1, pp. 55–68, 2013.

[25] Y. Wu, Z. Wei, and R. H. Deng, "Attribute-based access to scalable media in cloud-assisted content sharing networks," IEEE TMM, vol. 15, no. 4, pp. 778–788, 2013.

[26] Y. Zheng, X. Yuan, X. Wang, J. Jiang, C. Wang, and X. Gui, "Enabling encrypted cloud media center with secure deduplication," in Proc. Of AISACCS, 2015.

[27] S. Lederer, C. M ¨ uller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in Proc. of the 3rd ACM Multimedia Systems Conference, 2012.

[28] R. Pagh and F. Rodler, "Cuckoo hashing," J. Algorithms, vol. 51, no. 2, pp. 122–144, 2004.

[29] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang et al., "Named data networking," ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 66–73,2014.

[30] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," in Proc. of ACM HotNets, 2011.

[31] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in Proc. of IEEE INFOCOM, 2012.

[32] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in Proc. of NDSS, 2014.

[33] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. of ACM CCS, 2006.

[34] Z. V. Kilroy Hughes, David Singer, "ISO/IEC 23001-7 3nd Edition - Common encryption in ISO base media file format files," on line at: http://mpeg.chiariglione.org/standards/mpeg-b/common-encryption-iso-base-media-file-format-files, 2014.

[35] Matt Burns, "Watch out netflix, amazon instant video reaches 100,000 on-demand titles," online at: http://techcrunch.com/, 2011.

[36] N. T. Spring and D. Wetherall, "A protocol-independent technique for eliminating redundant network traffic," ACM SIGCOMM Computer Communication Review, vol. 30, no. 4, pp. 87–95, 2000.

[37] A. Anand, V. Sekar, and A. Akella, "Smartre: an architecture for coordinated network-wide redundancy elimination," in ACM SIGCOMM Computer Communication Review, vol. 39, no. 4, 2009, pp. 87–98.