**Abstract**

Mobile devices are widely used for uploading/downloading media files such as video, audio and images to/from remote servers. These devices, on the other hand, have limited resources and require offloading of some media processing tasks to the clouds for further processing. Migration of these tasks means that the media services provided by the clouds must be authentic and trusted by the mobile users. The existing schemes for secure exchange of media files between the mobile devices and the clouds have limitations in terms of memory support, processing load, battery power, and data size. These schemes lack the support for large-sized video files and are not suitable for resource-constrained mobile devices. This paper proposes a lightweight and efficient security scheme. The proposed scheme considers High Efficiency Video Coding (HEVC) Intra encoded video streams in unsliced mode as a source for data hiding. The main emphasize in proposed scheme is to support real-time processing with power saving constraint. Well-known Advanced Encryption Standard (AES) technique in 256 bit version is used as a base. The simulation results clearly show that the proposed scheme outperforms AES-256 by rising the processing time up to 4.76% and data size up to 0.72% approximately. The proposed scheme can readily be applied to real-time cloud media streaming.

**Key Words:** Cryptography, Cloud Computing, Mobile devices, HEVC, Intra, Unsliced Mode, AES, Authentication

1. **Introduction**

Mobile communication plays a vital role in our daily lives and has seen an unprecedented technological growth over the last two decades. Apart from using mobile devices for basic telecommunication services such as messaging and voice, an increasing number of users are using their phones for sharing multimedia data by taking pictures and video recording. As a result, a huge amount of data is generated by mobile users all the time. According to latest Cisco report, it grew up to 69% in 2014 with 2.5 exabytes per month at the end of 2014, whereas 4G connections produced 10 times more data traffic than non-4G connections, as shown in **Figure 1** [**1**]. The column represents data in units of exabytes while row represents years.



Figure 1. Cisco forecast for mobile data traffic [1]

Although, the storage capability of mobile devices has had a huge improvement, however, it still cannot meet the ever-increasing storage requirements of mobile users. Cloud platform provides a facility to mobile users to store their data remotely in media clouds and retrieve later at any point of time, thus, providing an ease for mobile users with an option to enhance their virtual storage [**2**], as shown in **Figure 2**. Accessing data over wireless channels from media clouds has become very popular among mobile users, primarily due to the development of various multimedia applications [**3**]. However, transmission of private data over the wireless links is prone to various security breaches. As a result, security has become a major concern for the data residing over media clouds [**4**]. As the data can be transferred and stored in a cloud system through wireless links, it is vulnerable to alteration, unauthorized disclosure and replay attacks. Trust of the users need to be guaranteed while uploading their multimedia and secret data over clouds. Although, there are existing approaches, which provide a facility to share secret information among a number of authorized users but are applied on simple form of data [**5-10**]. However, in case of videos, especially the large-sized videos, such approaches become restricted by data size and the processing capability of end user devices. As a result, lightweight but secured schemes need to be designed in this context.

The growing popularity of High Definition (HD) videos and the emergence of HD and beyond-HD formats such as, 4k×2k or 8k×4k resolutions are creating a benchmark for visual



Figure 2. Mobile data storage

The growing popularity of High Definition (HD) videos and the emergence of HD and beyond-HD formats such as, 4k×2k or 8k×4k resolutions are creating a benchmark for visual quality [**11-13**].The High Efficiency Video Coding (HEVC) is an emerging standard to deal with such high resolution multimedia contents, proposed by ITU-T and ISO/IEC [**14**]. This standard provides three basic modes for encoding multimedia contents, i.e., Intra, low delay and random. The intra mode treats each video frame as an independent image and focuses on quality rather than compression. This mode contains only I-frames and is suitable for such applications, where there is no compromise on visual quality such as video surveillance and live high-level video conferences. The low delay mode provides more compression with a compromise on visual quality and provides almost 33% reductions in bit rate [**15**]. This mode is a combination ofI and P-frames and is suitable for such online applications where compromise on visual quality can be made, such as online gaming, which is usually played on mobile devices with low resolutions. The random mode is a combination of I, P and B-frames with a facility to access any part of any frame. This mode offers better compression as compared to the low delay mode and is mainly used for contents storage with almost31% reductions in bit rate [**15**].

As a result of the above distinguishing features of HEVC standard, researchers are focused on developing security techniques which may utilize the encoded contents, produced by HEVC standard. Such techniques not only require an analysis of HEVC-produced contents for data protection but also require authorization by key exchanging as shown in **Figure 3**. Security is of utmost importance especially when such videos are uploaded to public clouds and later on retrieved by the users. Various encryption algorithms are proposed for HEVC standard to protect the video sequences against unauthorized attacks by malicious users who use third-party tools and methods to steal the transmitted video sequences [**16, 17**]. However, these algorithms focus only to restrict the video access but do not provide any facility to secure secret data hidden in such videos. Also, the specific requirements of mobile users such as processing capability, hardware resources and power backup are ignored while downloading and uploading their multimedia data to public cloud. Our research has the following main contributions.

1. The HD videos always contain redundant information. If these videos are encoded in Intra domain, the compression is too low for maintaining the visual quality. As a result of low compression, there is always abundant of space available for hiding secret information. We propose a cryptographic algorithm by using Intra encoded HD video sequences in public clouds. Though it introduces a slight overhead in nominating a specific video frame from input video sequence but provides a major benefit of enough space to hide secret data.
2. The secret data is shared in a public cloud which is semi trusted and can’t be fully hacked. Therefore, if the intruder downloads the video and steals the public key, it will only be able to crack half of the data, which means the decryption process can’t be completed without being an authentic user, having private key. Moreover, the targeted users are mobile users, which can be at different geographic locations across the globe. Our proposed approach supports mobility of users by authenticating them remotely and independently.
3. Our proposed approach is a combination of both public and private keys and doesn’t require continuous synchronization among users, as shown in **Figure 4**. The only responsibility of uploading user is to encrypt and upload the data and then broadcast a message to all authorized users that a new data has been uploaded. Afterwards, it is choice of other users whether they would like to download and decrypt the data or not. By possessing the same private key, they may download and fully decrypt the data at any time.
4. The proposed approach works efficiently in public clouds by providing the required security and reduces the cost of private cloud. Our proposed approach efficiently utilizes the computing power of cloud resources by partially (half) decrypting the data, thus minimizing the decryption processing delay at receiver side. As the targeted users are mobile users, it is not necessary that they would have enough computing and power resources.

As the main purpose of public clouds is to provide enough storage space and computing power to the users, therefore, the cloud environment is perfect for HD videos which demand large storage and computational efficiency. Our proposed scheme increases the compressed video size up to 0.72% on average as compared to other state-of-art techniques which increase the size up to 9% or higher approximately [**18**]. Our approach affects the size of transmitted binary video data but that variation is quite small and can easily be ignored as compared to other state-of-the-art approaches. However, this variation totally depends on the quantity of encrypted data. If the quantity is not that large, the increase in the size of binary video data would not be that high and would be very difficult to diagnose unlike other approaches, where the increase is of quite high. Although, we targeted the HD videos but our proposed approach is equally applicable to any other resolution of videos.

The rest of the paper is organized as follows. In Section 2, related work is presented. In 3, our proposed approach is explained in detailed, followed by simulation setup and results in Section 4. Finally, we conclude the paper in Section 5.



Figure 3. Traditional security scheme



Figure 4. Updated security scheme

1. **Related Work**

Mobile devices such as, smartphones and tablets are becoming an indispensable part of our lives for entertainment and convenient communication. With the increasing popularity of various mobile devices, there is a phenomenal growth in the development of mobile applications such as, email, web browsing, mobile games, terrestrial navigation, mobile healthcare, and social networking [**19-23**]. These applications indicate that mobile devices such as smartphones and tablet PCs are quickly becoming the dominant computing platforms for provisioning of seamless connectivity and entertainment regardless of user mobility. Mobile devices on the other hand are still limited in terms of their resources such as computational capabilities, storage and battery life time [**24**]. Moreover, they have limited communication resources such as available bandwidth and connectivity [**25, 26**]. The resource-constrained nature of these devices limits the support for developing various mobile applications. Mobile Cloud Computing (MCC) has addressed the resource-constrained drawback associated with mobile devices. MCC allows the mobile devices to offload the computationally intensive and space-demanding tasks to the cloud [**27, 28**]. Clouds, on the other hand, have ample of resources and as such provide an ideal platform for resource-consuming mobile applications such as speech recognition and video decoding [**29, 30**]. A number of such applications can be offloaded to the clouds for processing and utilizing the huge amount of resources of these clouds. Computation offloading of these applications save energy and improve the performance of mobile applications [**31, 32**]. MCC also enables mobile users to store/access large amount of their data on the cloud via wireless networks. This characteristic of MCC enables the mobile devices to save their data storage capacity and processing power [**33**].

The on-demand nature of cloud computing faces various security threats such as data loss, data leakage, denial of service, account or service traffic hijacking, and malicious insiders. A malevolent hacker may delete target’s data or the data may be lost due to a careless cloud service provider. To tackle this challenge, the important data needs to be encrypted and the encryption keys must be protected [**34, 35**]. If an intruder gains access to a customer credentials stored on the cloud, it may eavesdrop on the transactions and activities, maliciously manipulate the data, return falsified information, and redirect the customer to illicit sites. Denial of service is another concern for cloud platforms because the organizations are dependent on the 24/7 availability of one or more services [**34**]. The denial of one or more services may cost the service providers to lose their customers and may prove costly to customers as well especially when they are billed based on disk space consumed and compute cycles. Account or service hijacking is another major threat faced by cloud platforms [**35**]. Hijacking a service allows a malicious person to sneak into crucial and sensitive areas of a deployed service which may lead to breaching the integrity, availability and confidentiality of such service. Last but not the least, a malicious insider which can be a current or former employee, a business partner or a contractor, may gain access to the data, network or system for malicious purposes [**36, 37**]. The situation gets worse if the cloud service provider is solely responsible for data security.

Cloud platforms attract more attacks due to their distributed nature [**38**]. It is desirable that the data (video content in this context) is protected and may only be accessed in encrypted form. Hiding data directly in encrypted HEVC video streams to intact its quality can avoid the leakage of video content, which can address the security and privacy concerns associated with cloud computing [**39-42**]. A cloud server has the ability to embed the additional information about a video such as video notation and authentication data, into an encrypted version of HEVC format [**16**]. Once the data is hidden, the server can verify the integrity without knowing the original content. As a result, security and privacy of the encrypted data is preserved. In literature, various studies exist on encrypting the data in HEVC format [**16, 17**]. A novel encryption method for Intra and Inter frames in MPEG videos is presented in [**40, 44**]**.** The authors argued that highly sensitive and private videos require encrypting the video as a whole. Therefore, not only the Intra frame, but the Inter frames are also required to be encrypted. In [**45]**, an encryption scheme for MPEG videos is proposed. The proposed scheme is based on Advanced Encryption Standard-128 bit (AES-128) algorithm. Only Intra frames of a particular video are encrypted because the Inter frames are useless without knowing the corresponding Intra frames associated with these Inter frames. In [**46]**, a novel scheme based on selective encryption for HEVC data is proposed. The proposed scheme ensures transparent encryption and protection against various attacks. The proposed encryption and decryption is quite fast at the time of preserving the formation and length of video streams. Various clouds such as Google App Engine and Amazon web services have experienced various security attacks during recent years [**47**]. These security flaws are exploited by illegal users to steal either secret information or disturb the normal operation of Internet. As a result, robust and lightweight authentication and authorization schemes are required for the devices (mobile users in this case) interacting with the cloud platforms. Cloud computing is a variant of client-server architecture model, where, thousands of clients use the same infrastructure at a much larger scale. Identity and access control management is a core requirement for cloud computing **[48-50]**. Therefore, stronger authentication than conventional client-server inter-networking is required. In **[51]**, the authors proposed a public key and mobile out of band-based authentication for data stored on a cloud platform. Their proposed scheme transmits data in plaintext form which can easily be intercepted by malevolent entities. Moreover, the proposed scheme does not take into account the user privacy, data confidentiality and data integrity. Hence, their scheme is not suitable for a real-time cloud computing platform. Bilinear pairing in an elliptic curve has recently gained attention in developing an ID-based cryptosystem **[52-54]**. This cryptosystem solves the high cost issue of authentication and public key management derived from traditional public key cryptosystems. Here, the identity of a user is used as the public key of this user. Therefore, a user does not require extra computational cost for verifying the public keys of other users. Moreover, no extra storage space in the user’s device is required to store public keys of others and their corresponding certificates. Several studies have recently applied ID-based cryptosystems in various cloud environments. In **[55]**, a new ID-based authentication scheme is designed for cloud environment. Although, the proposed scheme is suitable for a distributed mobile cloud services environment, however, it lacks the support for user anonymity and un-traceability. Most of the authentication schemes which are based on elliptic curve or bilinear pairing **[52-54]** are designed for client–server environment. They are not feasible to be directly applied into distributed services environment where multiple service providers compete with each other and offer various types of services. The user needs to manage multiple private keys learned from each service provider. To resolve this issue, the simplest way is that all service providers share the same master private key. However, if an adversary acquire this master private key, it can poses as any one of the service providers in order to cheat the users. Moreover, an intruder who captures the master private key may learn the session keys as well. After learning the session key, the attacker can get sensitive information transmitted between another service provider and a user.

1. **Proposed Approach**

In this section, we present our proposed data hiding scheme, which is a combination of both public and private keys. The base algorithm is Advanced Encryption Standard (AES) in 256 bits version [**56**]. Our proposed scheme is directly applied on HEVC encoded HD stream. This scheme has three main phases in general, i.e., HEVC video encoding and data encryption, half decryption and full decryption with data extraction. The uploading user or data owner first encodes the video and then encrypts the data with encryption key to generate an encrypted HEVC Encoded Video Stream (HEVS). Secondly, the owner uploads the encrypted video over a nominated public cloud, for example, Amazon, where the cloud sources half decrypt the data without figuring out what is hidden in video stream and what are the contents of video stream. The receiver or downloading user downloads the encrypted video, decrypts it, extracts the required data and then either keeps or discards the video stream. Figure 5 and 6 show the overall process of proposed scheme.



Figure 5. Encoding and encryption



Figure 6. Decryption and/or decoding

1. **Video Encoding and Data Encryption**

The HD videos usually contain N number of frames, where Nϵ{500, 600, 700…}. Firstly, for real-time processing, it is not necessary to process and encode an entire video sequence. Secondly, HEVC codec and HD videos require enough hardware resources and computing power. As our targeted users can be mobile users, therefore, the above two conditions should be considered before starting encoding process. An easy solution to solve these limitations is by reducing the total number of frames during encoding. Alongside encoding limitations, it is not compulsory to encrypt entire HEVS. The encryption of entire HEVS increases the computational and time cost which also affects the format of HEVS. Therefore, the solution for these later limitations is to nominate a part of HEVS for encryption purpose on random bases to increase the security level. To keep the format of HEVS undisturbed, it is quite safe to encrypt secret data in Spatial Information (SI). Also the Motion Vectors (MVs) information can be utilized for this purpose [**57**].

In this paper, HEVS-based encryption scheme is proposed which is time-efficient, less complex in terms of computations and does not disturb the format of HEVS. After encoding the HD videos using the standard settings in HEVC, three main parts in HEVS can easily be utilized to encode the secret data, i.e., SI, MVs and Intra Prediction Modes Information (IPMI). As compared to [**57**], our proposed scheme encrypts the secret data in compressed domain, not in encoding domain. This feature makes it very easy to directly apply this scheme and modify the HEVS. Modified versions of pure AES have already been applied on H.264/AVC codec based encoded video streams [**16, 58**]. The common point between our proposed scheme and the schemes presented in [**16, 58**] is the use of AES for data encryption purpose. The first major difference between our proposed scheme and [**16, 58**] is that we are using HEVC/H.265 codec which is the latest one and they have used H.264 which is the previous one. Secondly, we are using HD videos, which are high resolution and difficult to deal with while, they have used low resolution videos. In this paper, we have used AES-256 by involving Public Key (PUK) and Data Encryption Key (DEK) along with cloud computing to make it more secure and less computational complex. We encrypt the secret data by distributing it in three parts, i.e., first part in SI, second part in MVs and the third part in IPMI in order to make the compressed HEVS format consistent. After encryption, the HEVS still can be decoded perfectly with any latest version of HEVC codec without showing any visible difference in decoded videos. The procedure to hide secret data in nominated sections of HEVS is very complicated and requires deep knowledge of HEVC encoder. A slight mistake can disturb the HEVS, which ultimately alerts the decoder and causes decoder crash and issues format mismatch errors [**16**].

* + 1. **Encryption in Spatial Information**

To increase the security level, the secret data is divided equally in three parts. The first part is encrypted in SI. As the HD videos are encoded in Intra domain, which means lots of spatial information, is available to utilize. This phase summarizes the embedding of secret data in SI. Both in H.264 and HEVC/2.65 standards, the entropy coding is basically used to encode the quantized coefficient values of SI [**59**]. The main difference is that in HEVC, only CABAC version is used while in H.264, CABAC and CALVC both are used. The details of CABAC framework can be found in [**59, 60**]. The CABAC framework has three main parts, i.e., binarization, context modeling and Binary Arithmetic Coding (BAC). We skip the detail explanation of these three parts, as it is out of scope of this paper. Secondly, we encode the HD videos in Intra domain, where the data compression is very less. As a result, it does not fully utilize the CABAC and the only focus is on BAC but at minor level.

The BAC is based on the rule of Least Probable Symbol (LPS) and Most Probable Symbol (MPS), which is adopted from Shannon-Fano coding [**69**]. The calculation of LPS and MPS is based on probability by dividing the possibility of occurrence of a symbol within a predefined interval, usually denoted by R. The first half of R, i.e., [0 0.5] reserves the probability of LPS while [0.6 1] half reserves the probability of MPS. The interval between [0.5 0.6] is considered as neutral and the symbol occurring in this interval range can either be treated as LPS or MPS. Based on the rule of LPS and MPS, in our proposed scheme, we utilize each byte of zero video data to hide the secret data. The secret data is first converted into binary format and then converted into bytes format. Secondly, a 2’s compliment operation is performed on secret data bytes. Thirdly, the negated secret data bytes are added with zero video bytes. In that way, the format and the number of bits in video data byte will remain in the same order and the original pattern can easily be extracted back at receiver side by converting those modified video data bytes into zeros using simple subtraction operation.

* + 1. **Encryption in Motion Vector**

The MVs play an important role in video coding, especially in Inter and Random access domains. In case of Intra domain, blocks of pixels are used to reference other blocks but within the same frame. This type of referencing is known as Intra prediction. The MVs are basically used to track moving regions and objects. Based on that motion information, compression and information discarding procedure is performed in video coding. They also help to keep the texture information of different objects in a video frame. The HEVC codec can encode MVs with much better precision and less residual errors. The main reason behind this accuracy is the Intra prediction directions which are 35 in number while there are only 9, present in H.264 standard. As stated above, only CABAC version of entropy coding is available in HEVC, therefore, CABAC is used to encode calculated MVs for transmission purpose [**61**].

After encoding through CABAC, the MVs transform into codewords. Each codeword is a combination of binary 1s and 0s and its length increases with increasing number of MVs. The structure of codewords is very simple. The most significant bit(s) in codewords is always zero. The length of codewords can easily be calculated by following equation:

**(1)**

where CL represents codeword length, AZ represents number of appended zeros and Bs represents the total number of bits, used to represent the binary presentation of decimal value. The decimal value indicates the MV number. The length of AZ is always 1 less than the Bs, i.e.

**(2)**

where LAZ and LBs represent total length of appended zeros and length of total number of bits respectively.

For example, if a MV is represented by decimal number 13, then its equivalent binary representation is 1101. As this binary presentation contains 4 bits, therefore, 3 zeros will be appended at most significant side. As a result, the total number of bits will become 7 in number, which is the length of codeword, used to represent that specific MV. It is clear now that the most significant part of a codeword is always zero, therefore, it can easily be utilized to encrypt second part of hidden data. Again the same procedure is performed here to encrypt the secret data, as followed in previous section to hide first part, which again will maintain the order and format of codewords.

* + 1. **Encryption of Intra Prediction Mode Information**

Unlike H.264 standard, which supports only three types of Intra coding modes, i.e., Intra\_4×4, Intra\_8×8 and Intra\_16×16 [**62**], there are N number of Intra coding modes in HEVC, from Intra\_2×2 to Intra\_64×64 [**63**]. To perform compression and prediction in H.264, the video frames are divided into equal size blocks, known as macroblocks (MBs). To provide more accuracy, the concept of MBs is replaced by Coding Tree Unit (CTU) in HEVC [**59**]. To make the encoding process fast and simple, Intra\_32×32 and Intra\_64×64 block sizes are selected. The IPMI is always available in the header of selected block size. The header also specifies the patterns, used to perform encoding process. The IPMI is encoded through CABAC, which transforms this information into codewords. Again the codewords creation follows the same criteria, as stated previously. The noticeable feature in these codewords is that the encoding patterns in consecutive blocks are always same for luma and chroma components [**70**]. Therefore, two options are available for encrypting the third part of secret data. Either the most significant zero bits are used or one of the coding pattern in consecutive blocks is used. To make it a standard, we use the most significant zeros to embed the third and last part of secret data by applying the same procedure, as followed before.

* 1. **Half Decryption**

As stated before, the data owner encrypts and uploads the data on the cloud. The application system on cloud provides three major services, i.e., Secure Storage Space (SSS), Key Generator (KG) and Half Decryption Function (HDF). In order to run such application system over public clouds, few assumptions are made:

* The main purpose behind the concept of public clouds is that they should be used freely with a trust factor. Though, it is very obvious that cloud owner may see the uploaded contents but it cannot change them. Therefore, it becomes a responsibility of encryption scheme to secure the data in such environments.
* A user plays dual role, i.e., a user may upload and download the data. Therefore, a user may be a requester or owner.
* Each user generates same PRK but may generate different PUKs.
* To protect the data transmission between user and cloud over communication channels, it is assumed that it is protected by some security protocol, such as SSH.

As public cloud cannot be trusted fully, therefore, the application system running over the cloud may generate PUKs but it cannot determine the PRKs, held by the users. As the data owner uses PUK generated by cloud KG, therefore, cloud may be able to partially decrypt the data by using the same PUK. This partial decryption reduces the processing overhead at data receiving side and this is the responsibility of HDF. To make this system functioning properly, each user registers itself with KG by forwarding user ID and user generated PUK. The KG replies by forwarding a newly generated PUK to generate the DEK.

* 1. **Full Decryption with Data Extraction**

This phase is very simple. When a user requests for a specific data, it sends the required data request to KG. The KG performs two checks. First, it filters out the authenticated user list to match the user ID. If the user ID does not match, it simply drops the connection. If the match gets successful, then the KG checks whether the requested data is available in the SSS or not. If the requested data is not available, it informs the requester about the absence of data, otherwise sends back the half decrypted data back to the requester. After getting the encrypted data, user decrypts the data fully by using its PRK. After extracting the secret data, it is up to the requester desire whether to keep the video stream for further use or simply discards it.

There can be two scenarios for data decryption. In the first scenario, the user is authorized one. In such situation, the maximum probability is that it will first decrypt the encrypted video stream to extract the secret data and then will decide either to keep or discard it. In second scenario, the user is unauthorized one but somehow has stolen the authorized user ID to access the encrypted video. In this scenario, the unauthorized user will be able to decode the video stream but will be unable to extract the hidden information without having the PRK.

* 1. **Proposed Algorithm**

The step by step explanation for proposed encryption and decryption scheme is as follows:

**Step 1: Registration Phase**

1. Each Mobile User (MUi) generate a PRKi and PUKi, where i=1,2,…n.
   * 1. MUi is either data owner or requester
2. Each MUi forwards its PUKi along with user ID to KG function on cloud.
3. KG authorizes MUi for uploading/downloading data by forwarding Specific Partial Key (SPK) and KG’s generated One Public Key (OPK).
4. OPK is a linear combination of MUn-1 PUKs.
5. The data owner generates Data Encryption Key (DEK), a linear combination of OPK and user-specific PRK.

**Step 2: Encoding Phase**

1. The videos are encoded through HEVC standard in Intra mode.
2. Output of this phase is a Binary File (BF), also known as HEVS.

**Step 3: Calculation Phase**

1. Calculates total number of blocks (TBs) in generated BF.
2. Separate the DC/zero Blocks (DCBs) and AC/non-zero value blocks (ACBs).

**Step 4: Pre-Encryption Phase**

1. Set up the total Number of Users (NOUs) and Number of Shadows (NOSs).
2. Using Shamir Secret (SS) scheme, the value of *r*in SS is set as NOSs-1.

**Step 5: Encryption Phase**

1. Input: Video frames or Binary Packets
2. Firstly, the information of SI, MV and IPMI is extracted from frames header.
3. Frame headers are distributed in two groups: One contains true shadows and second contains false shadows.
4. Secondly, AES-256 is applied on those headers, nominated for having true shadows to hide the secret data.

**Step 6: Uploading Phase**

1. After getting authorized, data owner (MUi) alerts KG function that data is ready to upload.
2. KG function acknowledges MUi and starts receiving the data.
3. After receiving the complete data file, KG stores it in SS.

**Step 7: Half Decryption Phase**

1. This step involves cloud computing resources.
2. The encrypted data will partially be decrypted at the cloud side to reduce the user processing load by HDF.
3. The PUK generated by KG is used to perform this task.

**Step 8: Downloading/decryption Phase**

1. MUi is authenticated.
2. After authentication, MUi downloads the required video, decrypts and/or decodes it by using the PRK to extract the secret data.
3. **Simulation Setup and Results**

In this section, we present the experimental setup and results of our proposed scheme. Our proposed scheme is mainly based on modified version of AES-256 but we also compare the results of our scheme with other versions of AES and DES. Though AES-128, AES-192 and DES-56 are also very popular but to assure better security, large sized security keys are always preferable and difficult to crack. **Table 1** shows a summarized comparison among AES, 3DES and DES techniques.

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristics** | **AES** | **3DES** | **DES** |
| Length | 128, 192 and 256 bits | 112 and 168 bits | 56 bits |
| Type | Symmetric | Symmetric | Symmetric |
| Resistance | Defensive against linear, differential and interpolation attacks | Vulnerable against differential attack | Vulnerable against linear and differential attacks |
| Keys Production | 2128, 2192 and 2256 | 2112 and 2168 | 256 |
| Approximated cracking time to check all possible keys | 2128: 5x1021 years  2192: 7.5x1032 years  2256: 10x1042 years | 2112: 800 days  2168: 1200 days | 256: 400 days |

Table 1. Comparison between AES and DES

In order to secure the secret key(s), Shamir’s Scheme (SS) [**64**] is adopted, which is based on (r, n) threshold scheme. This will distribute the authority of private key among *n* number of users. The reason to adopt SS is its unique feature to distribute the secret information into *n* number of shadows or parts, where *n* is equal to the number of users in a shared group, who are sharing the information with each other [**4**]. The experiments are performed on Dell machine with processor Intel ® CoreTM i5-3470 CPU @ 3.20 GHz and having memory 8 GB. For encoding purpose, we use HEVC/H.265 reference software version 16.6 with Intra configuration settings [**65**]. For simulation dataset, we use standard HD videos [**66**]. We select video sequences (PeopleOnStreet, Kimono, Cactus and Rush\_Hour) having resolution 2560×1600 and 1920×1080. The frame rate of test videos varies from video to video. **Table 2** shows the details of test sequences used during the experiments. As shown in Table 2, the test sequences have different frame rate. Similarly, these sequences contain multiple objects, moving with different velocities. For testing purpose, we encode first 150 frames from each video sequence. The Group of Pictures (GOP) is set as 1 because of Intra coding mode [**59**]. The proposed technique is applied only on Intra-frames (I-frames). The main reason behind applying it only on I-frames is that in case of I-frames, the compression is always less. Therefore, we have enough space available in videos frames to hide our required information/data. Though on the other side, it can also be applied to Inter-frames (P or B-frames) but this increases computational time of encryption and decryption, as Inter-frames are always dependent on I-frames. Another main problem for applying the proposed scheme on Inter-frames is that if an I-frame is lost during transmission due to any reason, all dependent Inter-frames will also get dropped which will cause failure in the delivery of required information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Sequence** | **Total Frames** | **Encoded Frames** | **Frame Rate** | **Resolution** |
| PeopleOnStreet | 150 | 150 | 30 | 2560×1600 |
| Kimono | 240 | 150 | 24 | 1920×1080 |
| Cactus | 500 | 150 | 50 | 1920×1080 |
| Rush\_Hour | 500 | 150 | 25 | 1920×1080 |

Table 2. Details of test video sequences

The encryption algorithm is implemented in Matlab version R2015a. For cloud and cloud-based application implementation, again we use same version of Matlab. The current version of Matlab provides Amazon EC2 cloud facility with an ease for the programmer to use built in functions to perform cloud based computing [**67**]. As our main target is real-time processing, therefore, for comparison purpose, our main focus is on processing time and size of the test video sequences. The computational time and size changes in encoded videos are also noted after applying proposed scheme, AES-256, AES-192, AES-128 and DES. **Table 3** shows the details of encoded sequences bit rate, encoding time and sizes before and after applying encryption. Some comparison words are required here. Performance graph in **Figure 7** is based on the calculations of Table 3. It is very clear from Table 3 and Figure 7 that the proposed scheme almost keeps the same size of encoded video stream, as it is before encryption while the other schemes produce significant change. The average computational time [**68**] required to apply the above mentioned encryption techniques are shown in **Table 04**. Performance graph in **Figure 8**is based on the calculations of Table 4. As our proposed scheme is based on AES-256, therefore, it is very clear from Table 4 and Figure 8 that it produces slightly less time as compared to AES-256 but increased time as compared to DES, AES-128 and AES-192 because of large keys lengths.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Sequence** | **Bit Rate (Kbps)** | **Size after encoding and before encryption (MB)** | **Encoding time (Sec)** | **Size after applying DES (MB)** | **Size after applying AES-128 (MB)** | **Size after applying AES-192 (MB)** | **Size after applying AES-256 (MB)** | **Size after applying Proposed Scheme (MB)** |
| PeopleOnStreet | 121122.57 | 45.701606 | 22805.77 | 49.35773448 | 50.14119058 | 50.14459058 | 50.24149058 | 45.856471 |
| Kimono | 23406.06 | 18.285984 | 10113.745 | 20.48030208 | 18.72484762 | 18.7248566 | 18.73454522 | 18.719000 |
| Cactus | 156169.819 | 28.563682 | 12900.051 | 31.99132384 | 36.56151296 | 36.56182233 | 36.57183695 | 28.564881 |
| Rush\_Hour | 64912.965 | 30.713254 | 12225.882 | 30.95896003 | 16.46230414 | 16.46545433 | 16.47321188 | 30.756332 |

Table 3. Time and size comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Sequence** | **Computational Time of DES (Sec)** | **Computational Time of AES-128 (Sec)** | **Computational Time of AES-192 (Sec)** | **Computational Time of AES-256 (Sec)** | **Computational Time of Proposed Scheme (Sec)** |
| PeopleOnStreet | 2.141594 | 0.749083 | 0.859941 | 0.947595 | 0.913500 |
| Kimono | 0.856887 | 0.299721 | 0.344077 | 0.379149 | 0.342021 |
| Cactus | 1.338504 | 0.468181 | 0.537466 | 0.592251 | 0.580066 |
| Rush\_Hour | 1.439234 | 0.503413 | 0.577914 | 0.636821 | 0.612243 |

Table 4. Computational time comparison

Figure 7. Size comparison

Figure 8. Time comparison

**Conclusion**

Hiding the secret data in encrypted video streams is a new era of research which has started grabbing attention of researchers because of privacy and security issues in public clouds. In this article, a security scheme is presented which hides the secret data in HEVC encoded video stream, i.e., in compressed domain. The scheme consists of three major phases, i.e. video encoding, data encryption and decryption with/without decoding. The proposed scheme almost maintains the original video stream size after encryption without affecting the visual quality of video data, thus making it an ideal platform for real-time video applications. As the secret data is distributed in encoded video stream, thus it makes it very difficult for hackers to extract complete secret data, as they do not know the exact locations and patterns of hiding scheme. Another major advantage is that our proposed scheme fully supports the encoding and decoding structure of HEVC standard. The encrypted video stream can perfectly be decoded without getting corrupted or showing any sign of extra hidden information. Experiments have shown that the proposed scheme maintains the visual quality with a slight compromise on increasing encoded video stream, which is totally dependent on the quantity of secret data, needs to be hidden.

**References**

1. “Global Mobile Data Traffic Forecast Update 2014-2019”, White Paper, Cisco Visual Networking, February 2015.
2. Y. Wen, X. Zhu, J. Rodrigues and C. Chen, “Cloud Mobile Media: Reflections and Outlook”, IEEE Transactions on Multimedia, Vol. 16, No. 04, June 2014.
3. Z. Yin, F. Yu, S. Bu and Z. Han, “Joint Cloud and Wireless Networks Operations in Mobile Cloud Computing Environments with Telecom Operator Cloud”, IEEE Transactions on Wireless Communications, Vol. 14, No.07, March 2015.
4. H. Wang, S. Wu, M. Chen and W. Wang, “Security Protection between Users and the Mobile Media Cloud”, IEEE Communications Magazine, Vol. 52, No. 03, March 2014.
5. L. Ferretti, F. Pierazzi, M. Colajanni and M. Marchetti, “Performance and Cost Evaluation of an Adaptive Encryption Architecture for Cloud Databases”, IEEE Transactions on Cloud Computing, Vol. 02, No. 02, June 2014.
6. L. Ferretti, M. Colajanni and M. Marchetti, “Distributed, Concurrent and Independent Access to Encrypted Cloud Databases”, IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 02, February 2014.
7. K. Yang and X. Jia, “Expressive, Efficient and Revocable Data Access Control for Multi-Authority Cloud Storage”, IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 07, July 2014.
8. H. Liu, H. Ning, Q. Xiong and L. Yang, “Shared Authority based Privacy-Preserving Authentication Protocol in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 01, January 2015.
9. Q. Tang, “Nothing is for Free: Security in Searching Shared and Encrypted Data”, IEEE Transactions on Information Forensics and Security, Vol. 09, No. 11, November 2014.
10. R. Padilha and F. Pedone, “Confidentiality in the Cloud”, IEEE Security and Privacy, Vol. 13, No. 01, February 2015.
11. K. Weide and D. Levitas, “Assessing the High-Definition Online Video Opportunity”, White Paper, April 2009.
12. S. Tanwir and H. Perros, “A Survey of VBR Traffic Models”, IEEE Communications Surveys and Tutorials, Vol. 15, No. 04, January 2013.
13. “Understanding Ultra High Definition Television”, Ericsson White Paper, November 2015.
14. HEVC, “High Efficiency Video Coding (HEVC) Text Specification Draft 6”, Technical Report, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, USA, 2012.
15. M. Karczewicz, “Video Coding Technology Proposal by Qualcomm”, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-A121, Joint Collaborative Team on Video Coding (JCT-VC), Dresden, Germany, April 2010.
16. Z. Shahid and W. Puech, “Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings”, IEEE Transactions on Multimedia, Vol. 16, No. 01, January 2014.
17. A. Boho, G. Wallendael, A. Dooms and R. Walle, “End-to-End Security for Video Distribution”, IEEE Signal Processing Magazine, Vol. 30, No. 02, March 2013.
18. C. Paar and J. Pelzl, “Introduction to Public Key Cryptography”, A Textbook for Students and Practitioners, Springer, 2009.
19. S. Lim, P. Bentely, N. Kanakam, F. Ishikawa and S. Honiden, “Investigating Country Differences in Mobile App User Behavior and Challenges for Software Engineering”, IEEE Transactions on Software Engineering, Vol. 41, No. 01, September 2014.
20. T. Dunsmuir, B. Payne, G. Cloete, C. Petersen and J. Ansermino, “Development of mHealth Applications for Pre-Eclampsia Triage”, IEEE Journal of Biomedical and Health Informatics, Vol. 18, No. 06, January 2014.
21. X. Hu, X. Li, N. Ngai, V. Leung and P. Kruchten, “Multidimensional Context-Aware Social Network Architecture for Mobile Crwodsensing”, IEEE Communications Magazine, Vol. 52, No. 06, June 2014.
22. A. Rahmati, C. Shepard, C. Tossell, L. Zhong and P. Kortum, “Practical Context Awareness: Measuring and Utilizing the Context Dependency of Mobile Usage”, IEEE Transactions on Mobile Computing, Vol. 14, No. 09, October 2014.
23. Y. Choi, C. Yoon, Y. Kim, S. Heo and J. Silvester, “The Impact of Application Signaling Traffic on Public Land Mobile Networks”, IEEE Communications Magazine, Vol. 52, No. 01, January 2014.
24. L. Liu, W. Zhu, S. Yin, E. Tang and P. Peng, “An Uneven-Dual-Core Processor Based Mobile Platform for Facilitating the Collaboration Among Various Embedded Electronic Devices”, IEEE Transactions on Consumer Electronics, Vol. 60, No. 01, February 2014.
25. A. Celebi, M. Kenkel and T. Wong, “Bandwidth Enhancement of a Compact Transverse Bilateral Helical Antenna With Parasitic Element for Mobile Device Applications”, IEEE Transactions on Antennas and Propagation, Vol. 63, No. 03, January 2015.
26. L. Liu, Y. Yi, J. Chamberland and J. Zhang, “Energy-Efficient Power Allocation for Delay-Sensitive Multimedia Traffic Over Wireless Systems”, IEEE Transactions on Vehicular Technology, Vol. 63, No. 05, March 2014.
27. S. Barbarossa, S. Sardellitti and P. Lorenzo, “Communicating While Computing: Distributed Mobile Cloud Computing Over 5G Heterogeneous Networks”, IEEE Signal Processing Magazine, Vol. 31, No. 06, November 2014.
28. Y. Zhang, D. Niyato and P. Wang, “Offloading in Mobile Cloudlet Systems with Intermittent Connectivity”, IEEE Transactions on Mobile Computing, Vol. 14, No. 12, February 2015.
29. M. Assefi, M. Wittie and A. Knight, “Impact of Network Performance on Cloud Speech Recognition”, 24th International Conference on Computer Communication and Networks, pp: 1-6, August 2015.
30. C. Lai, Y. Lai, M. Wang and J. Niu, “An Adaptive Energy-Efficient Stream Decoding System for Cloud Multimedia Network on Multicore Architectures”, IEEE Systems Journal, Vol. 08, No. 01, October 2013.
31. X. Chen, L. Jiao, W. Li and X. Fu, “Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing”, IEEE/ACM Transactions on Networking, Vol. PP, No. 99, October 2015.
32. K. Elgazzar, P. Martin and H. Hassaanein, “Cloud-Assisted Computation Offloading to Support Mobile Services”, IEEE Transactions on Cloud Computing, Vol. PP, No. 99, August 2014.
33. C. Chen, M. Won, R. Stoleru and G. Xie, “Energy-Efficient Fault-Tolerant Data Storage and Processing in Mobile Cloud”, IEEE Transactions on Cloud Computing, Vol. 03, No. 01, July 2014.
34. M. Ficco and F. Palmieri, “Introducing Fraudulent Energy Consumption in Cloud Infrastructures: A New Generation of Denial-of-Service Attacks”, IEEE Systems Journal, Vol. PP, No. 99, April 2015.
35. B. Wang, H. Li, X. Liu, F. Li and X. Li, “Efficient Public Verification on the Integrity of Multi-Owner Data in the Cloud”, Journal of Communications and Networks, Vol. 16, No. 06, December 2014.
36. K. Choo, “A Cloud Security Risk-Management Strategy”, IEEE Cloud Computing, Vol. 01, No. 02, July 2014.
37. J. Ryoo, S. Rizvi, W. Aiken and J. Kissell, “Cloud Security Auditing: Challenges and Emerging Approaches”, IEEE Security and Privacy, Vol. 12, No. 06, November 2013.
38. Q. Yan and F. Yu, “Distributed Denial of Service Attacks in Software-Defined Networking with Cloud Computing”, IEEE Communications Magazine, Vol. 53, No. 04, April 2015.
39. K. Liu, M. Au, W. Susilo, K. Liang, R. Lu and B. Srinivasan, “Secure Sharing and Searching for Real-Time Video Data in Mobile Cloud”, IEEE Network, Vol. 29, No. 02, April 2015.
40. B. Zeng, S. Yeung, S. Zhu and M. Gabbouj, “Perceptual Encryption of H.264 Videos: Embedding Sign-Flips into the Integer-Based Transforms”, IEEE Transactions on Information Forensic and Security, Vol. 09, No. 02, December 2013.
41. U. Puvvadi, K. Benedetto, A. Patil, K. Kang and Y. Park, “Cost-Effective Security Support in Real-Time Video Surveillance”, IEEE Transactions on Industrial Informatics, Vol. 11, No. 06, October 2015.
42. H. Shen, L. Zhou and Y. Zhao, “An Efficient Motion Reference Structure Based Selective Encryption Algorithm for H.264 Videos”, IET Information Security, Vol. 08, No. 03, May 2014.
43. S. Li, G. Chen, A. Cheung, B. Bhargava and K. Lo, “On the Design of Perceptual MPEG-Video Encryption Algorithms”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 02, February 2007.
44. D. Xu, R. Wang and Y. Shi, “Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution”, IEEE Transactions on Information Forensic and Security, Vol. 09, No. 04, January 2014.
45. Z. Shahid, M. Chaumont and W. Puech, “Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I and P Frames”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 21, No. 05, March 2011.
46. H. Hofbauer, A. Uhl and A. Unterweger, “Transparent Encryption for HEVC Using Bit-Stream-Based Selective Coefficient Sign Encryption”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp: 1986–1990, May 2014.
47. J. Correa and J. Ricaurte, “Web Application Development Technologies Using Google Web Toolkit and Google App Engine-Java”, IEEE Latin America Transactions, Vol. 12, No. 02, March 2014.
48. T. Jung, X. Li, Z. Wan and M. Wan, “Control Cloud Data Access Privilege and Anonymity with Fully Anonymous Attribute-Based Encryption”, IEEE Transactions on Information Forensic and Security, Vol. 10, No. 01, November 2014.
49. K. Liu, M. Au, X. Huang, R. Lu and J. Li, “Fine-Grained Two-Factor Access Control for Web-Based Cloud Computing Services”, IEEE Transactions on Information Forensic and Security, Vol. PP, No. 99, October 2015.
50. S. Ruj, M. Stojmenovic and A. Nayak, “Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds”, IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 02, February 2013.
51. Z. Le, N. Xiong, B. Yang and Y. Zhou, “SC-OA: A Secure and Efficient Scheme for Origin Authentication of Interdomain Routing in Cloud Computing Networks”, IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp: 243-254, May 2011.
52. F. Guo, Y. Mu, W. Susilo, H. Hsing, D. Wong and V. Varadharajan, “Optimized Identity-Based Encryption from Bilinear Pairing for Lightweight Devices”, IEEE Transactions on Dependable and Secure Computing, Vol. PP, No. 99, June 2015.
53. J. Tsai, “A New Efficient Certificateless Short Signature Scheme Using Bilinear Pairing”, IEEE Systems Journal, Vol. PP, No. 99, November 2015.
54. R. Azarderakhsh, D. Fishbein, G. Grewal, S. Hu and R. Verma, “Fast Software Implementations of Bilinear Pairings”, IEEE Transactions on Dependable and Secure Computing, Vol. PP, No. 99, December 2015.
55. J. Yang, Y. Chang and C. Huang, “A User Authentication Scheme on Multi-Server Environments for Cloud Computing”, 9th International Conference on Information, Communications and Signal Processing (ICICS), pp: 1-4, December 2013.
56. A. Bogdanov, D. Khovratovich and C. Rechberger, “Biclique Cryptanalysis of the Full AES”, Advances in Cryptology, Lecture Notes in Computer Science, Vol. 7073, pp: 344-371, 2011.
57. G. Wallendael, A. Boho, J. Cock and A. Munteanu, “Encryption for High Efficiency Video Coding with Video Adaptation Capabilities”, IEEE Transactions on Multimedia, Vol. 59, No. 03, August 2013.
58. M. Asghar and M. Ghanbari, “An Efficient Security System for CABAC Bin-Strings of H.254/SVC”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 23, No. 03, March 2013.
59. J. Sullivan, J. Ohm, H. Woo-Jin and T. Wiegand, “Overview of the High efficiency Video Coding (HEVC) Standard”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, September 2012.
60. D. Marpe, H. Schwarz and T. Wiegand, “Context-Based Binary Arithmetic Coding in the H.264/AVC Video Compression Standard”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 07, July 2003.
61. V. Sze, M. Budagavi and G. Sullivan, “High Efficiency Video Coding (HEVC): Algorithms and Architectures”, Springer, 2014.
62. B. Juurlink, M. Mesa, C. Chi, A. Azevedo and C. Meenderinck, “Understanding the Application: An Overview of the H.264 Standard”, Scalable Parallel Programming Applied to H.264/AVC Decoding, Briefs in Computer Science, pp: 5-15, Springer, May 2012.
63. J. Lainema, F. Bossen, W. Han, J. Min and K. Uger, “Intra Coding of the HEVC Standard”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, December 2012.
64. Adi Shamir, “How to Share a Secret”, Communications of the ACM, pp: 612-613, 1979.
65. Bossen F.: Common HM test conditions and software referenceconfigurations,’’ document JCTVC-L1100 of JCT-VC, Geneva,CH (2013).
66. <https://media.xiph.org/video/derf/>
67. <http://www.mathworks.com/products/parallel-computing/parallel-computing-on-the-cloud/>
68. <http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/>
69. <https://en.wikipedia.org/wiki/Shannon%E2%80%93Fano_coding>
70. C. Poynton, “Digital Video and HDTV: Algorithms and Interfaces”, Morgan-Kaufmann, 2003.