Journal of Computer and Communications, 2014, 2, 18-25
Published Online October 2014 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2014.212003

Improved High Definition Multimedia Interface Authentication Mechanism

R. N. Iyare[1], S. D. Walker[2]

[1]Department of Physics and Electronics, Adekunle Ajasin University, Akungba Akoko, Ondo State, Nigeria
[2]School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
Email: rachel.iyare@hotmail.com, stuwal@essex.ac.uk

Received 20 August 2014; revised 18 September 2014; accepted 8 October 2014

Abstract

Security information has become so significant in transmission due to the rapid advancement in digital data exchange. Thus, it is necessary to protect the confidentiality and licensing of video content from illegal access. Currently, High-bandwidth Digital Content Protection (HDCP) provides the confidentiality and licensing of digital content for High Digital Multimedia Interface (HDMI). In this paper, we have been able to show how cryptanalysts have conducted attacks on the HDCP protocol showing its vulnerability in protecting digital contents. Therefore, the HDCP scheme is seriously flawed and compromised. Encryption and decryption of audio/video files were implemented in both Data Encryption Standard (DES) and Advanced Encryption Standard (AES) algorithms by means of Python Cryptography Toolkit (Pycrypto). Security analysis such as key space analysis and statistical analysis were carried out on the encrypted image. The result of the analysis indicates that AES algorithm is secure and robust; hence the authentication and licensing of HDMI can be improved by implementing HDCP design with AES algorithm.

Keywords

Security, Encryption, Cryptanalyst, Authentication

1. Introduction

The means of transporting high-definition (HD) video, multi-channel HD audio and 3D (three dimensional) encoded HD video is of great concern to customers. Customers derive a great pleasure from home entertainment system; hence they would go to any length to get the best from their equipment. Currently, it is of great importance to use high quality connectors between devices that are optimized for signal transfer in order to get the true benefit of the improvements in video products. Amongst all the audio/video (AV) interconnects that are availa-

How to cite this paper: Iyare, R.N. and Walker, S.D. (2014) Improved High Definition Multimedia Interface Authentication Mechanism. *Journal of Computer and Communications*, **2**, 18-25. http://dx.doi.org/10.4236/jcc.2014.212003

ble, the patronage for HDMI is more as it delivers the best picture, high-definition television (HDTV) and supports AV cabling over the same cable. As a result of this more and more consumer electronic devices are made with HDMI connectivity and a wide range of HDMI devices are currently in use.

The issue of content protection is as a result of the introduction of the HDMI standard which is used in digital communication of high definition content. Content providers involved in the traditional analogue communication methods experienced problems in analogue hole in which it allowed attackers to gain access to content without any restriction. Content providers utilized the opportunity of the introduction of a new standard to eradicate the analogue hole, and hence Intel Cooperation decided to include a content protection scheme as part of the new standard [1] [2].

The licensing and confidentiality of video content transmitted over HDMI is currently made available by HDCP which is a specification developed by Intel to protect digital content across HDMI [3]. HDCP is the encryption system used in HDMI and DVI to protect content from being copied by unauthorized users. Every single device possesses a unique set of keys that are generated by the HDCP licensing authority and dispensed in a manner that any two devices can create a shared key which is used to encrypt content shared between two devices through the use of a unique set of keys [4].

However, the approach of this HDCP to content security is seriously flawed and compromised. Rob Johnson in [5] implemented an attack on the HDCP protocol by borrowing a number of TV screens that contained device keys so as to decrypt HDCP signals. The master key of the HDCP was built by means of some powerful 128 computers. However, the reconstructed master key was discovered to be the same as the one popped online two weeks later. The two keys were proficient to generate keys that could encrypt and decrypt HDCP even though they had different minor representations [6].

Cryptanalysts in [7] also discovered several weaknesses in the HDCP scheme and the master key that was used by the licensing authority was made public on the internet. As a result, the HDCP key exchange protocol has been rendered inefficient since anyone can use the master secret to produce a device that is capable of mimicking any other HDCP device.

The main contribution of this paper is to show how cryptanalyst can be used to reveal the vulnerability of HDCP in protecting digital contents. In addition, the paper also highlights how the authentication of HDMI can be improved upon by using AES encryption algorithm in the design of HDCP.

The remainder of the paper is organised as follows. Section 2 describes the HDCP protocol and attacks conducted by cryptanalyst showing its vulnerability. Section 3 presents the methodology. Finally Sections 4 and 5 show the results and conclusion respectively.

2. High-Bandwidth Digital Content Protection

In the digital world, each copy is basically a perfect copy of the original; as a result there is increased dependence upon copy protection in order to secure content and to avoid content compromise from unauthorized users. Content providers have become more concerned about illegal copying and use of digital content due to its extensive accessibility hence content protection technologies have been deployed by content providers, electronic manufacturers and media manufacturers to protect access to high-value content delivered through dissimilar media [8]. One of the links that is exposed to attacks in utilizing HDMI is the digital baseband interface between receiving a set-top box (STB) and high-definition television (HDTV) display device [8]. Hence an encryption technology known as the HDCP is used in HDMI to protect content from being copied by unauthorized users [9]. HDCP was developed by Intel Cooperation in 1999 to protect encrypted digital content [10], prevent pirates from capturing digital content that is transmitted over the digital visual interface (DVI) bus connecting DVD players and computers to video monitors [11]. In an all-digital content distribution framework the DVI bus could be the most susceptible link without HDCP hence pirates could get around the encryption on Blu-ray Disks and DVDs by playing the content over the DVI bus and recording the unencrypted, uncompressed exact digital video stream to afterward playback later [12]. As shown in **Figure 1**, HDCP interface protects high-value content as it travels between HDCP transmitters and receivers.

The HDCP Authentication Protocol

Crosby and other cryptanalysts in [7] were able to explain the version of HDCP that apprehended the cryptographic portions of HDCP, in that the Key Selection Vector (KSV) is assigned to a public vector v_A

$E\left(Z/2^{56}Z\right)^{40}$ and a private vector, u_A $E\left(Z/2^{56}Z\right)^{40}$. The vector u_A was kept in tamper-proof hardware but in the event of software implementation it was through code obfuscation techniques. Also, whenever devices *A* and *B* wished to communicate, they interchanged their vectors v_A and v_B respectively, thereby *B* computing $K' = v_A$. u_B and *A* computing $K = v_B \cdot u_A$ as shown in **Table 1**. In order to authenticate if the key agreement process has been successful in the HDCP, the receiver responds with the 16-bit value that is computed by $R' = h\left(K', n_A\right)$ when the transmitter has sent a nonce n_A. Johnson [5] also reasoned along with Crosby that HDCP devices comprises of 40-dimentional bit vectors over $Z/2^{56}$. Similarly, Johnson and Crosby have the same view about the HDCP authentication protocol as shown in **Figure 2** except that the HDCP specification of Johnson does not describe how the public and private keys were generated.

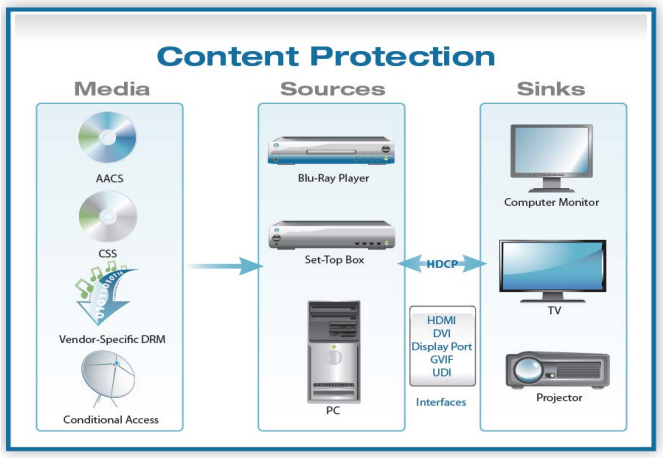

Figure 1. The content protection chain; Source: [8].

Table 1. HDCP protocol variables; Source: [7].

Name	Size	Comment
v_A, v_B	40 bits	Must have Hamming weight 20
u_A, u_B	Vector of 40 56-bit numbers	
n_A	64 bits	
K, K'	56 bits	$K = v_B \cdot u_A$ $K' = v_A \cdot u_B$
R, R'	16 bits	$R = h\left(K, n_A\right)$, $R' = h\left(K', n_A\right)$

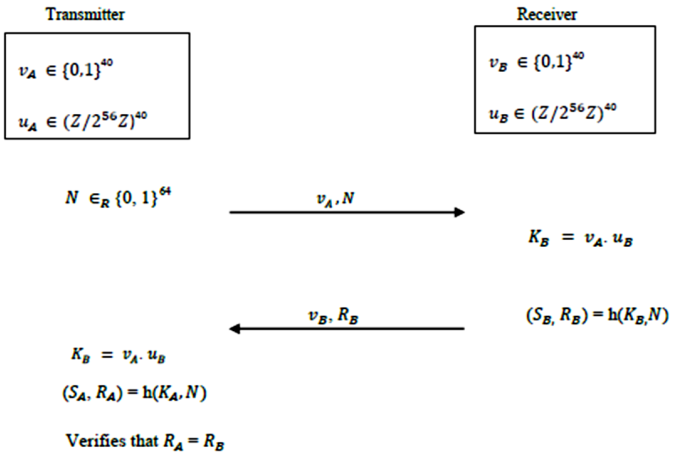

Figure 2. A simple version of the HDCP protocol; Source: [7].

3. Methodology

Since HDMI transfers uncompressed digital audio/video data, uncompressed audio and video files of 36 Kb and 20,471 Kb respectively were used in the Python Cryptography Toolkit which is a stable and dependable base for writing python programs. The following steps were taken in encrypting and decrypting the audio and video files:

1. Objects such as operating system (os), random and struct were imported from Crypto. Cipher derived from AES library.
2. Next was the declaration of encrypt File function together with parameters.
3. There are several ciphering mode that is allowed in block cipher symmetric algorithms [13] and the most detectable one is Electronic CodeBook (ECB). This mode has a security weakness due to the fact that the same data is being ciphered to the same value [14] [15] [16] as a result selected Cipher Block Chaining mode (CBC) was selected in our implementation (CBC provides improved security as every single block is dependent on the preceding file) and the block mode of AES was set. The encryption method operates with input consisting of 16 bytes blocks and Pycrypto encryption application programming interface (API) is low-level.
4. By means of using Crypto.Cipher.AES new, a new object known as AES encryptor object was created before the encryption mode and key were assigned to the object.
5. A conditional statement was declared in the program such that if there is no output file, the encryptor will create one using the input file name as well as its extension.
6. The Pycrypto block level encryption operates with key length of 16, 24 or 32 bytes long. In this paper, 32 bytes key length was used since encryption strength increases proportionally with the length of the key
7. For data encryption, the encryptor generates a random value known as initialization vector (IV), this is passed into the encryptor object and then the IV is randomly generated for each new encryption so as to prevent attackers from cracking the encryption. Also, the initialization vector is implemented together with the secret key so that the encryption will be very complicated for hackers and this is also aimed at to preventing data encryption repetition [17].
8. The size of the file and the struct object was written to the output file and the input file was read according to the chunk (mass of the data) size.
9. Also, a condition was declared such that, if the input file is empty, the code execution should break.
10. Finally, the file was encrypted through the use of the encryptor function.

Subsequently, to decrypt the encrypted file, the size of the original file is read first from the 32 bytes constituting the encrypted file and then to initialize the object (AES object), the initialization vector is read. The file is therefore decrypted into piece (chunks) *i.e.*, truncated thereby obtaining the actual size of the file.

Implementation of DES

The same uncompressed digital 36 Kb audio file and 20,471 Kb video file were used respectively in the Python Cryptography Toolkit. The following steps were taken to encrypt and decrypt the audio/video files:

1. Objects such as operating system (os), random and struct were imported from Crypto. Cipher derived from DES library.
2. Next was the declaration of encrypt File function together with parameters.
3. By means of using Crypto.Cipher.DES.new, DES encryptor object was created then the encryption mode and key was assigned to this object.
4. A conditional statement was also declared in the program such that if there is no output file, the encryptor will create one with input file name plus extension.
5. DES uses a key length of 8 byte therefore in this encryption 8 byte key length was also used.
6. The initialization vector (IV) was passed into the encryptor object and the IV was randomly generated for each new encryption.
7. The size of the file and the struct object were written into the output file and the input file was read according to the chunk size.
8. Check if the input file is empty, terminate the code execution.
9. Finally, the file was encrypted through the use of the encryptor function.

However, to decrypt the audio/video files in DES algorithm, this involved the cipher block which is made up of the secret key, the audio/video file and the algorithm. The decryptor throws a condition that if the input file is empty, then the code execution should break. Finally the encrypted file was decrypted by means of the decryptor

function.

4. Output Analysis

The audio and video files were encrypted and decrypted for AES and DES by means of a PyCryptor Cryptography Toolkit in Python programming language. After encrypting the 36 Kb uncompressed audio file by means of a secret key in DES encryption algorithm, the encrypted audio file could not play since it was encrypted and then decrypting the encrypted audio file, the audio file started playing as the original file. We also observed that after encrypting 20,471 Kb video file in the same encryption algorithm, the encrypted video file could not play but after decryption the full video file was restored.

In addition, since AES is usually employed in hardware implementation as it is a very fast symmetric block algorithm [18]-[21] and protects very sensitive data we therefore deployed AES encryption algorithm both for the audio and video files. The encrypted audio file did not play after encryption until it was decrypted with the same secret key used for encryption. After encryption of the video file in this same algorithm, the volume of the encrypted video was then 55 Kb (representing just the audio being that it was an audio/video file) although the video stream was still encoded. But after decrypting the encrypted avi, the full audio/video stream of 20,471 Kb was restored.

4.1. Security Analysis

4.1.1. Statistical Analysis

With the objective of frustrating powerful attacks based on statistical analysis, two methods of confusion and diffusion were proposed by Shannon [22]. The superior diffusion and confusion properties of AES have been demonstrated by performing statistical analysis on it in [23]. The ciphered image in the AES algorithm was observed to be rather identical and appreciably different from the unencrypted image in **Figure 3**. And most assuredly, after implementing the encryption in **Figure 4** and decryption stages, there was no loss of any kind in the quality of the image and the decrypted file was the same in magnitude with the original image in **Figure 3**. This observation was the same for the audio file in **Figure 5**, all in AES algorithm; the size of the original audio was 36

Figure 3. Original image (before encryption); Source: wildlife [24].

Figure 4. Encrypted image; After encryption.

Kb, after encryption and decryption depicted in **Figure 6** and **Figure 7** reveal that the size of the audio was still the same.

4.1.2. Key Space Analysis

In this work key space size of 128 bit was used and this is large enough to oppose every form of brute-force attacks. During the encryption of this image using different keys, AES was observed to be very sensitive to the secret key. In the encryption code, the output of the encryption was structured to be in the form of ciphertext not encrypted

Figure 5. Audio (before encryption).

Figure 6. Encrypted audio (after encryption).

Figure 7. Decrypted audio (after decryption).

file. After using different secret key for six consecutive times, completely different ciphertext was observed in all the secret keys used hence the encrypted image is different for these keys compared to the encrypted image when the original secret key was used. AES algorithm therefore, guarantees security as it possesses a feature of sensitivity to different keys during the encryption process.

5. Conclusions and Further Work

In the implementation of the HDCP protocol, discovery has shown that the standard used for the HDCP design involves a DES encryption algorithm which has a key length of 56 bits, and the actual key space in the design being less than 40 as each key space has 20 bits set to 1 and 20 bits set to 0. Hence, through the use of special hardware it can be easily cracked by attackers.

The strength of encryption is not necessarily measured in bits or key size, but it is determined by the encryption algorithm design. HDMI being the major focus of this work transfers uncompressed digital audio/video data, so encryption and decryption of audio/video files were performed by means of AES encryption algorithm due to the fact that it is a robust algorithm that is still unbreakable, and thus it is not susceptible to brute-force attacks.

Furthermore some security analysis was carried out on the AES encrypted file showing that AES algorithm protects the confidentiality and integrity of sensitive information and AES algorithm is also made up of 128-bit data length which is large enough since encryption strength is stronger with larger key length. Further works can be done on the HDCP protocol by implementing its design with AES algorithm since this encryption algorithm is more secure and robust. Through the use of this encryption algorithm, it is possible that the HDCP will be very difficult for attackers to crack.

Acknowledgements

We acknowledge the technical support of the members of the telecommunication research group, University of Essex. We also appreciate Dr. Norbert Volker for his technical advice while preparing the manuscript.

References

[1] Bright, P. (2012) Claimed HDCP Master Key Leak Could Be Fatal to DRM Scheme. Cited: 16 July 2012, Online. http://arstechnica.com/tech-policy/2010/09/claimed-hdcp-master-key-leak-could-be-fatal-to-drm-scheme/

[2] HDCP Technologies-Digital Content Protection, LLC. (2012) Cited: 28 July 2012, Online. http://www.digital-cp.com/hdcp_technologies

[3] Westerkamp, D. (2007) HDMI and HDCP—The Manufacturer's Perspective. EICTA Article.

[4] Evain, J. (2012) HDCP—The FTA Broadcasters' Perspective. Cited: 25 July 2012, Online. http://www.ebu.ch/fr/technical/trev/trev_312-eviain_hdcp.pdf

[5] Johnson, R., Rubnich, M. and DelaCruz, A. (2011) Implementing a Key Recovery Attack on the High-Bandwidth Digital Content Protection Protocol. *Proceedings of Consumer Communications and Networking Conference* (*CCNC*) *IEEE* 2011, Las Vegas, 9-12 January 2011.

[6] Roettgers, J. (2012) Scientists Release HDCP Decryption Tool. Cited: 24 July 2012, Online. http://gigaom.com/video/scientists-release-hdcp-decryption-tool/

[7] Crosby, S., *et al*. (2012) A Cryptanalysis of High Bandwidth Digital Content Protection System. *Proceedings of the* 2002 *ACM Digital Rights Management Workshop*, Alexandria, October 30-November 3, 2006, 192-200.

[8] HDCP Deciphered: Digital Content Protection LLC. White Paper, 2008.

[9] Candelore, B. and Diego, S. (2009) Wireless Video Communication. United States Patents: US2009/0103471 A1. April 23.

[10] Zhao, J.J., *et al*. (2011) On Weaknesses of the HDCP Authentication and Key Exchange Protocol and Its Repair. *Elsevier Journal*, 19-25.

[11] LLC Digital Content Protection (2008) High Bandwidth Digital Content Protection System: Interface Independent Adaptation.

[12] Stockfisch, M. (2007) HDMI/DVI HDCP Handshake Problems and How to Avoid Them. Quantum Data Inc. White Paper.

[13] Schneier, B. (1996) Applied Cryptography: Protocols, Algorithms and Source Code.

[14] Bourbakis, N. and Dollas, A. (2003) Scan-Based Compression-Encryption Hiding for Video on Demand. *IEEE Multimedia Magazine*, **10**, 79-87. http://dx.doi.org/10.1109/MMUL.2003.1218259

[15] Shiguo, L., Jinsheny, S. and Zhiquan, W. (2005) A Block Cipher Based for the Chaotic Standard Map. *Chaos, Solutions and Fractals*, **26**, 117-129.

[16] Li, S.J., Zheng, X., Mou, X.Q. and Cai, Y.L. (2002) Chaotic Encryption Scheme for Real Time Digital Video. *Proceedings of SPIE*, **4666**, 149-160.

[17] Rouse, M. (2012) What Is Initialization Vector (IV)? http://whatis.techtarget.com/definition/initialization-vector-IV

[18] Gaj, K. and Chodowiec, P. (2001) Fast Implementation and Fair Comparison of the Final Candidates for Advanced Encryption Standard Using Field Programmable Gate Arrays. *Proceedings of the* 2001 *Conference on Topics in Cryptology*: *The Cryptographer's Track at RSA CT-RSA* 2001, San Francisco, 8-12 April 2001, 84-99.

[19] Hodjat, A. and Verbauwhede, I. (2004) A 21.54 Gbits/s Fully Pipelined AES Processor on FPGA. *Processing of the* 12*th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, 20-23 April 2004, 308-309.

[20] Janvinen, K., Tominisko, M. and Skytta, J. (2003) A Fully Pipelined Memoryless 17, 8 Gpbs AES-128 Encryptor. *Proceedings of the International Symposium on Field Programmable Gate Arrays*, Monterey, 23-25 February 2003, 207-215.

[21] Mclone, M. and McCanny, J. (2003) Rijindael FPGA Implementations Utilizing Look-Up Tables. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, **34**, 261-275. http://dx.doi.org/10.1023/A:1023252403567

[22] Shannon, C.E. (1949) Communication Theory of Secrecy System. *Bell System Technical Journal*, **28**, 656-715.

[23] Zeghid, M., Machhout, M., Khriji, L., Baganne, A. and Tourki, R. (2007) A Modified AES Based Algorithm for Image Encryption. *World Academy of Science, Engineering and Technology*, **1**, 10. http://www.waset.org/journals/waset/v3/v3-86.pdf

[24] Wildlife Refuge by Golacula 153 Views-Wildlife.avi. August 2011. http://www.youtube.com/watch?v=ft9XGMd8I4c

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.