MULTIMEDIA CONTENT PROTECTION VIA BIOMETRICS-BASED ENCRYPTION

Umut Uludag and Anil K. Jain

Computer Science and Engineering Department, Michigan State University,
3115 Engineering Building, East Lansing, MI, 48824
{uludagum, jain}@cse.msu.edu

ABSTRACT

We propose a multimedia content protection framework that is based on biometric data of the users and a layered encryption/decryption scheme. Password-only encryption schemes are vulnerable to illegal key exchange problems. By using biometric data along with hardware identifiers as keys, it is possible to alleviate fraudulent usage of protected content. A combination of symmetric and asymmetric key systems is utilized for this purpose. The computational requirements and applicability of the proposed method are addressed. The results of encryption and decryption experiments related to time measurements are included. Watermarking systems can be used to complement the proposed method to permit novel uses of protected multimedia data.

1. INTRODUCTION

The utilization of digital techniques in the creation, editing and distribution of multimedia data offers a number of opportunities to a pirate user, such as high fidelity copying. Furthermore, the widespread usage of Internet is providing additional channels for a pirate to quickly and easily distribute the copyrighted digital content without the fear of being tracked. As a result, the protection of multimedia content (image, video, audio, etc.) is now receiving a substantial amount of attention.

Two of the most commonly used methods for protection of intellectual property rights (IPR) are digital watermarking and encryption. Dittmann *et al*. [1] discuss the applicability of these methods, security requirements of multimedia data and associated problems. Digital watermarking consists of embedding some information about the data (e.g., ownership) into the multimedia data itself. Hartung and Kutter [2] and Swanson *et al*. [3] provide an overview of watermarking techniques for different multimedia data. However, watermarking techniques are not robust to various attacks on multimedia data, e.g., filtering and cropping. Encryption can also be utilized to eliminate the problems of unauthorized copying and distribution. But, encryption suffers from the problem of illegal sharing of the keys as illustrated below. Suppose Alice has an encrypted multimedia file and let us assume that a pirate web site or a pirate user, Bob, is distributing this file. In order for Alice to use the file, she must also have the correct key to decode the data. Alice can obtain the key via legal means, e.g., by registering herself with the web site associated with the content and supplying her payment information. This way the content provider has the information about the user (Alice) who is about to view/play/listen (henceforth, this *utilization* of multimedia data will be referred collectively as *playing*) the protected content. However, Alice can also obtain the key via pirated means (e.g., Bob sends Alice the correct key, in addition to the encrypted file), which eliminates the security provided by encryption.

An additional source of information that can be embedded in the encryption process is related to the attributes of the physical system (hardware or software) utilized by the users. For example, the hard disk (HD) serial number, the operating system number, etc. can be used as keys in the encryption process. The decoder checks these numbers in a host computer and, if they are not the correct ones used during encryption, the data cannot be decoded correctly (Bob can not easily send his hard drive to Alice, and we assume that the encryption/decryption process can not be tampered, for example Alice can not tamper with her HD serial number). But a legitimate user may want to play the multimedia file in multiple systems, such as a notebook and a desktop computer. Using hardware identifiers in the encryption/ decryption processes eliminates such a possibility.

The Trusted Computing Platform Alliance (TCPA) and Palladium specifications involve encryption as a way to increase the security of the overall PC architecture [4], [5].

Another possible solution to illegal key exchange can be the use of biometric characteristics of the users, namely, their physiological or behavioral characteristics (e.g., fingerprints, face, iris) that are unique to an individual and hence can be used for personal authentication [6]. Unlike token-based authentication (e.g., ID cards or keys) and knowledge-based authentication (e.g., passwords and PINs), biometrics data cannot be easily forged or guessed [7], [8]. Assuming that the biometrics system is secure (for a thorough treatment of possible attacks on a biometrics system, see [9]), adding the biometric data of the user into the encryption/decryption process can increase the security of the digital content. For example, in the scenario mentioned earlier where Bob sent Alice a pirated file, now Alice would need to present Bob's finger to decode the pirated data! The possibility of using biometrics data in digital signature applications has been addressed in [10]. Janbandhu and Siyal [10] use biometric data (e.g., iris image) to create keys in asymmetric and symmetric key encryption systems. However, biometric signals of users are not invariant over time. For example, in the case of fingerprint images, these changes can occur because of the improper placement of the finger on the sensor, sensor noise, dry or dirty fingers and cuts and bruises on the fingers. Figures 1 and 2 show examples of this intra-class variability for fingerprint and iris images, respectively.

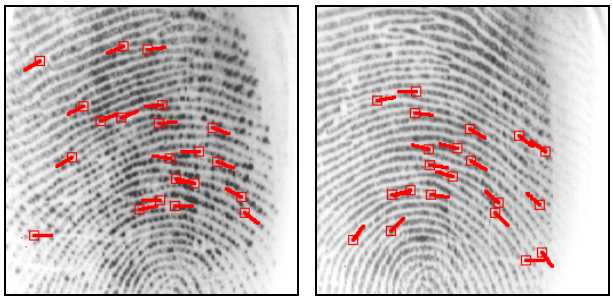

Figure 1. Two different fingerprint images of one user.

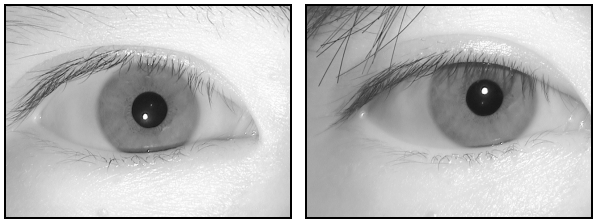

Figure 2. Two different iris images of one user.

The intra-class variations in the sensed biometric images lead to different biometric features (e.g., see Figure 1 that also shows the extracted minutiae overlaid on the fingerprint images). As a result, biometric data cannot be used directly to define a key in a digital signature system. Although these changes in biometric data are "small", the system becomes useless if the intra-class variability results in even a 1-bit change in the key. Note that in spite of these intra-class changes, the matcher will normally generate a higher score when the input is a pair of Bob's fingerprints, compared to the case when the input pair consists of one fingerprint from Bob and one from Alice. Janbandhu and Siyal [10] ignored this intra-class variability of the biometric data and assumed that the iris biometric is invariant for a user.

Soutar *et al.* [11] propose a mechanism in which biometric data of users are utilized to secure the keys in encryption/decryption processes, instead of protecting the keys via passwords. The method, which is called Biometric Encryption™ by the authors, functions as a key management system.

2. BIOMETRICS IN ENCRYPTION / DECRYPTION

We assume that there exist two communicating entities; the server S and the user U. U wants to receive the file V that resides at S. During file transfer, both asymmetric and symmetric key encryption schemes are used [12]. In Figure 3, K_S^+ and K_S^- denote the public and private keys of S, respectively. $K_S^+(.)$ and $K_S^-(.)$ denote the application of these keys in an asymmetric key system.

In the symmetric key system, $E(X, k_1, k_2, ..., k_n)$ denotes encrypting the file X first with key k_1, then encrypting the resulting file with k_2, and so on. Similarly, $D(Y, k_n, k_{n-1}, ..., k_1)$ denotes decrypting the file Y, first with key k_n, then decrypting the resulting file with key k_{n-1}, and so on. The data initially available at S and U are shown in respective columns inside dashed boxes in Figure 3. I_U is the identity of the user U, such as the user name. P_U is the password selected by the user to be used in encryption/decryption steps. B_U^t, $t = 0, 1, 2...$ denotes the biometric data of the user U, obtained at time t. Note that the biometric data is not invariant with respect to time due to the reasons cited before.

Now, we describe the encryption-decryption process in detail. First, using the public key of the server, user encrypts I_U, P_U and B_U^0, and sends this encrypted data to the server. The server decodes these three pieces of

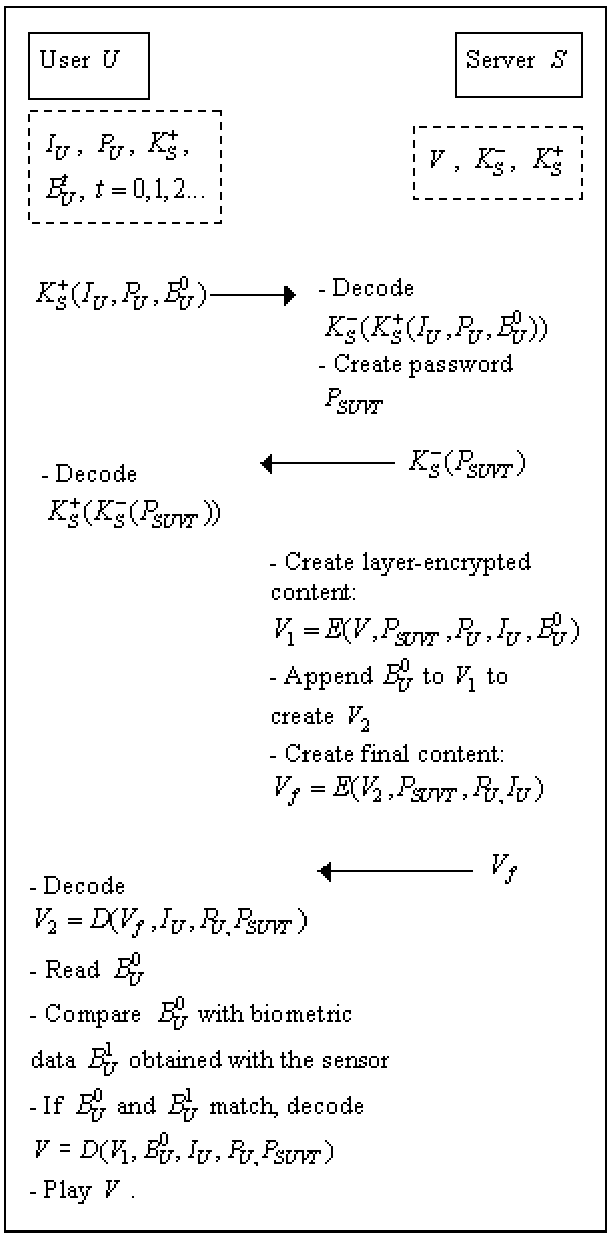

Figure 3. File transfer structure.

information by using its private key K_S^-. Due to this asymmetric key scheme, only server S, and not an intruder, can decode this information. After checking the validity of the user and related issues such as payment status, the server creates a password P_{SUVT} (which is a function of server S, user U, content V and a time stamp T) and sends it to the user after encrypting it with K_S^-. The user decodes this data by using public key K_S^+. The server creates the encrypted content V_1 by encrypting V in a layered manner with the keys generated from P_{SUVT}, P_U, I_U, and B_U^0. After appending the biometric data (B_U^0) to V_1, another layered encryption is carried out as shown in Figure 3 to arrive at V_f. The server sends this file to the user, where the keys used in encryption are used in reverse order to find V_2. Since this data (V_2) contains B_U^0, the biometric data obtained from the sensor, B_U^1, can be matched with B_U^0. If there is a positive match (i.e., B_U^0 and B_U^1 are from the same finger, iris, etc.), final layered decryption is carried out to arrive at the actual multimedia content V. This will enable the media player to play the content V for user U. Note that the next time the user wants to play the multimedia data, B_U^0 will be matched with B_U^2 and so on.

In the above secure file transfer scheme, we assume a "closed application", where the decrypted file is not stored at the user's computer but decrypted just before it is played. The biometric sensor, matcher, decryption module, media player and playing medium (e.g., monitor, speaker, etc.) are assumed to be connected together securely, where no tampering is possible.

3. COMPUTATIONAL REQUIREMENTS

When encryption and decryption are utilized in any system, the computational requirements become an important issue. Since the computational requirements of an asymmetric key system is several orders of magnitude larger than that of a symmetric key system, we use the former just for processing relatively small amounts of data, such as user identity, password, etc., and the latter for encrypting the multimedia data itself (which can be huge in size; for example a typical 3 min. music encoded in MP3 format can occupy 5 MB).

In the next section, we provide the times measured for typical encryption and decryption processes via the popular symmetric key system, Data Encryption Standard (DES) [13]. Also, Advanced Encryption Standard (AES) [14], another symmetric key algorithm that is the successor for DES, can be used for stronger security. Alternatively, cryptosystem architectures designed for multimedia data (e.g., [15]) can be used for reducing time complexity and increasing the applicability of the system, especially for real-time applications.

The utilization of biometric matching leads to two additional considerations. Due to the intra-class variations in the sensed biometric, every biometric system has some false rejects (conveyed via FRR or False Reject Rate) and some false accepts (conveyed via FAR or False Accept Rate). As an example, in a recent performance evaluation involving several fingerprint matching algorithms [16], the best algorithm had an Equal Error

Rate (EER), which is the point where FRR is equal to FAR, of 0.2%. Even though this FRR value (1 in 500 tries) is very small, it is still possible that a genuine user will not be accepted by fingerprint matcher. To eliminate this problem, the sensor may capture the same biometric several times, to check whether any of the captured biometric matches the template embedded in the multimedia file. Also, multiple biometric modalities (such as fingerprint, iris, etc.) can be used in encryption and decryption processes, and matching of any single biometric modality can suffice for initiating the decryption of the encrypted file. Another issue in using biometric data is the time needed for verifying a user. The FVC 2002 study [16] reported that the verification time for the best fingerprint matcher (with 0.2% EER) was 1.97 seconds. The above data suggest that fingerprint matching is viable for use in encryption/decryption processes to secure multimedia data as outlined in Figure 3.

4. EXPERIMENTAL RESULTS

We provide encryption and decryption times (wall clock times) for the application of DES on multimedia files. The standard key length in DES is 56 bits. Hence, the user ID (I_U), user selected password (P_U), and server generated password (P_{SUVT}) can be used directly as DES keys, assuming that these values are 8-character strings composed of 7-bit ASCII code. The biometric data (B_U^t , $t = 0,1,2...$) is generally larger in size; for example, a typical fingerprint image may generate a feature vector (composed of fingerprint minutiae location and orientation) that is more than 600 bits. Similarly, iris images can generate a feature vector with a 512-bit length. These feature vectors can be converted to 56-bit keys via one-way hash functions, and then utilized as DES keys.

Essentially, the encryption and decryption operations are very similar in DES, only the keys are utilized in reverse order. As a result, we can expect the encryption (at the server S) and decryption (at user U) times to be very similar for the same multimedia file. In fact, for a Sun Ultra 10 workstation (333 MHz), both encryption and decryption of a 5 MB file in the Cipher Block Chaining (CBC) mode of DES takes 3 seconds. Considering that the total number of DES decryptions required at user's computer is 7 (see Figure 3), the total decryption time for the file is around 21 seconds. This time is acceptable since the decryption is only carried out before playing the multimedia file. Furthermore, utilization of special hardware chips can reduce these times substantially [12].

5. CONCLUSIONS

A simple multimedia content protection scheme, which is based on layered encryption/decryption involving biometric matching, is proposed. The time required for the necessary encryption and decryption processes are provided for DES symmetric system; these times are acceptable. Furthermore, the utilization of special chips will reduce these times significantly. Hardware identifiers such as hard drive serial number can also be used in the encryption/decryption processes to bind the playing of multimedia data to the specific user equipment.

As a complement to the proposed architecture, watermarking and data hiding techniques can be utilized to address the requirements of novel uses of multimedia data, where editing of the multimedia content by the end user is allowed. A user may want to annotate the content, delete some parts of it, add a custom (audio, video) feature to it, filter it, etc. This necessitates allowing her to access the unencrypted file. If the biometric data of the end user can be embedded into the multimedia content, before encryption at the server, such that it is robust to these operations, biometric matching can be carried out even in the case of edited files. The utilization of secret keys associated with the hardware (and not known to the user) that is used in playing multimedia data can alleviate the piracy associated with capturing of data in the player (e.g., reading the display card, audio card buffers). But piracy is possible without even going into the digital domain; music can be captured with a microphone and a camcorder can record video that is being played. A possible solution to this problem is the use of *trusted* recorders, i.e., devices that do not record multimedia if a copyright identifier is present in the file, such as a hidden set of tones in audio.

6. REFERENCES

[1] J. Dittmann, P. Wohlmacher and K. Nahrstedt, "Using cryptographic and watermarking algorithms", *IEEE Multimedia*, vol. 8, no. 4, pp. 54-65, Oct-Dec. 2001.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques", *Proc. IEEE*, vol. 87, no. 7, pp. 1079-1107, July 1999.

[3] M.D. Swanson, M. Kobayashi and A.H. Tewfik, "Multimedia data-embedding and watermarking technologies", *Proc. IEEE*, vol. 86, no. 6, pp. 1064-1087, June 1998.

[4] *TCPA-Trusted Computing Platform Alliance*, http://www.trustedcomputing.org/tcpaasp4/index.asp

[5] A. Carroll, M. Juarez, J. Polk and T. Leininger, *Microsoft "Palladium": A Business Overview*, http://www.microsoft.com/PressPass/features/2002/jul02/0724palladiumwp.asp

[6] G. Lassmann, "Some results on robustness, security, and usability of biometric systems", *Proc. ICME 2002*, Lausanne, Switzerland, Aug. 2002, vol. 2, pp. 577-580.

[7] A.K. Jain, L. Hong, and S. Pankanti, "Biometric identification", *Comm. ACM*, vol. 43, no. 2, pp. 91-98, Feb. 2000.
[8] A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints", *Proc. IEEE*, vol. 85, no. 9, pp. 1365-1388, Sept. 1997.
[9] N.K. Ratha, J.H. Connell and R.M. Bolle, "An analysis of minutiae matching strength", *Proc. 3rd AVBPA*, Halmstad, Sweden, June 2001, pp. 223-228.
[10] P.K. Janbandhu and M.Y. Siyal, "Novel biometric digital signatures for Internet-based applications", *Inf. Management and Computer Security*, vol. 9, no. 5, pp. 205-212, 2001.
[11] C. Soutar, A. Stoianov, R. Gilroy and B.V.K.V. Kumar, *Biometric Encryption*,
http://www.bioscrypt.com/technology/white_papers.shtml
[12] B. Schneier, *Applied Cryptography*, Second Edition, John-Wiley, New York, 1996.
[13] National Institute of Standards and Technology, *Data Encryption Standard (DES)*, Federal Information Processing Standards Publication 46-2, 1993.
http://www.itl.nist.gov/fipspubs/fip46-2.htm
[14] National Institute of Standards and Technology, *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197, 2001.
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[15] C.P. Wu and C.C.J. Kuo, "Efficient multimedia encryption via entropy codec design", *Proc. SPIE*, vol. 4314, pp. 128-138.
[16] *FVC 2002, 2nd Int'l Fingerprint Verification Competition*,
http://bias.csr.unibo.it/fvc2002/results/resultsAvg.asp