Hindawi Publishing Corporation
Advances in Multimedia
Volume 2015, Article ID 956416, 5 pages
http://dx.doi.org/10.1155/2015/956416

Research Article

Security Requirements for Multimedia Archives

Sang Bae Park

KISTI, 245 Daehak-ro, Yuseong-gu, Daejeon 305-06, Republic of Korea

Correspondence should be addressed to Sang Bae Park; plucky@kisti.re.kr

Received 29 August 2014; Accepted 21 November 2014

Academic Editor: Seungmin Rho

With the explosive growth of various multimedia contents, digital archives are used to store those contents accordingly. In contrast to the traditional storage systems in which data lifetime is measured in months or years, data lifetime in the archive is measured in decades. This longevity of contents causes new security issues that threat the archive systems. In this paper, we discuss these new security issues in perspective. And we suggest some security requirements for digital archives.

1. Introduction

Now digital archives are increasingly being used to store digital contents that need to be preserved for a long time. Various kinds of multimedia contents, for example, cultural contents, academic journal, news, and so forth, are increasing very rapidly. Moreover, many existing contents are transferred into a digital multimedia format for availability and long-term preservation. Some digital contents should be preserved for a period of time. For example, the medical center has to store the patient's treatment for 10 years in republic of Korea. By the copyright law, copyright is effective for 70 years after the author's death. These kinds of contents should be carefully managed.

In order for the contents in such archives to be useful, there are some properties such as availability, integrity, and authenticity that should be protected. Generally, current storage systems adopt many security mechanisms, including access control, authentication, and encryption [1–3]. In contrast to traditional storage systems in which contents lifetimes are measured in months or possibly years, contents lifetimes in archive systems are measured in decades. For this longevity of contents, the security mechanisms for traditional storage system have some limitations. Primitives including block ciphers and hash functions do not guarantee robustness in the long term, it is a big challenge how to achieve long-term security [4–6].

In this paper, we discuss some security issues in digital archive systems and the plan for long-term preservations. Since a single cryptographic primitive cannot guarantee long-term security, we should consider additional physical protection methods and proper security policy. For example, write-once media can reduce the burden for authentication and integrity of stored contents. If we implement a hybrid-storage system with general storage and write-once media, we can store content in general storage and store log information in write-once media.

This paper is organized as follows. In Section 2, we briefly introduce a cryptographic background in information security. In Section 3, we discuss the long-term security issues, including the lifespan of cryptographic primitives, cryptographic keys, and other issues related to digital archive systems. In Section 4, we deal with required services for digital archives and update procedure for contents in archives.

2. Cryptographic Background

In this chapter, we briefly introduce some concepts and techniques for information security. For information security, we consider the following properties. Confidentiality guarantees that only authorized user can access the information. Authentication is for proving the originality. Integrity guarantees that there is no alteration. For satisfying these properties, we use physical protection, technical protection, and security policy. Cryptography is the technical approach for information security. Basic cryptographic primitives are symmetric key encryption, asymmetric key encryption, digital signature,

hash function, and PRNG (pseudo random number generator) [7].

Symmetric key encryption is for confidentiality. There are block ciphers and stream ciphers. Encryption makes plain text into cipher text with encryption key and decryption is its reverse process. Since encryption key and decryption key are the same, we call this function symmetric key encryption. For example, DES, AES, and RC4 are famous and are widely adopted [8, 9]. Symmetric key encryption is fast and efficient, but there is a problem how to share the key between message sender and receiver.

Asymmetric key encryption is also for confidentiality. Users make a pair of keys, public key and private key. The public key is used for encryption and the private key is for decryption. Since the public key is open to all users, key management for encryption is more convenient than symmetric key encryption. The most famous algorithm is RSA [10].

Hash function produces a fixed length random sequence for any input. This is for checking the integrity. MD4, MD5, and SHA-1 are famous hash functions [11–13]. Particularly, keyed hash function is called MAC (message authentication code).

A digital signature is an application of symmetric key encryption. Signer makes a hash value of the given message into a digital signature with the signer's private key. The verifier can check the authentication with the signer's public key. There are RSA and DSA [10, 14].

PRNG is used for making a random number. Unpredictability is most important for secure cryptographic protocols. Most cryptographic protocols including key management start from selecting random numbers.

Using these cryptographic primitives, people make cryptographic protocols for specific purposes. For confidence of the user's public key, we construct a TTP (trusted third party), the so-called CA (certificate authority). CA issues a digital signature for user's public key. People can check whether the user's public key is valid or not. This signature, including user's public key, is called a certificate. The PKI (public key infrastructure) includes CA and its service for secure application of asymmetric key cryptography.

Data enveloping is a practical approach for a content encryption. The sender makes a random session key and encrypts the session key with the receiver's public key. Then the sender encrypts main contents with the session key by symmetric key encryption. This approach has some advantages. At first, using symmetric key encryption for bulk data is efficient. Then the receiver can easily recover the whole message without a sharing process.

Another TTP service is a timestamp. This service is issuing a signature at that point of time. If the signer includes a timestamp in its signature, the verifier can check when the signature is generated.

For general storage service, there are some protocols. Since the size of stored data is growing more and more, transfer time became expensive. Proof of retrievability is for lessening this problem [15]. This protocol is that storage provider shows that user's data is really in the storage without transferring the whole data. Another protocol for storage service is a zero remnant protocol [16]. This protocol is verifying that there is no remnant after user's data deletion request.

3. Long-Term Security Issues

In this chapter, we discuss some security issues arising in digital archive systems. We consider the security of cryptographic primitives and related security protocols, lifetime of cryptographic keys, and other issues for long-term security.

3.1. Security of Cryptographic Primitives. Cryptographic primitives are basic tools for information security. Generally, we consider that cryptographic primitives are always secure. Though cryptographic primitives are secure now, these could be insecure in the future. There have been many researchers who try to find a new way to exploit cryptographic primitives' weakness.

In 1976, US government published the standard block cipher DES. DES had been used widely in many areas. At that time, 56-bit encryption key is sufficient. Because there was lack of computing power for 2^56 brute force, DES was considered secure. But, by the late 1990s, computers were so cheap and powerful that a 2^56 brute force search for the key became a feasible task. Moreover, dedicated cryptanalysis has been developed. Differential cryptanalysis and linear cryptanalysis are successful methods finding the key less than exhaustive search [17, 18].

Like block ciphers, hash functions are also mortal. The most famous hash functions MD-series, MD4, MD5, and MD5, are considered insecure, after Wang and Yu published collision for those hash functions that is a pair of different messages producing the same hash value [19].

The collisions of hash function cause a serious problem for digital signature using that hash function. Figure 1 shows how to make a fake message for a given digital signature. Collisions of hash functions nullify the authenticity of the digital signature. But there are still many private CAs adopting MD5 hash function. Certificates from those CA could be forged at a tolerable cost.

The security of cryptographic primitives determines the security of cryptographic protocols and services that consist of cryptographic primitives. For example, cipher suite of web security protocol SSL/TLS includes a cipher RC4 and hash functions MD4, MD5. The careless choosing of a cipher suite can cause the whole transaction insecure.

Table 1 shows a part of the known cryptanalysis for ciphers and hash functions.

Single cryptographic primitive might not guarantee long-term security. So we should continuously monitor the security of cryptographic primitives and we make a plan for algorithm change and contents update.

3.2. Lifetime of Cryptographic Keys. Cryptographic keys are most important in information security. There are two kinds of cryptographic keys. One is for a symmetric key cryptography, and the other is for an asymmetric key cryptography. Symmetric key cryptography includes symmetric

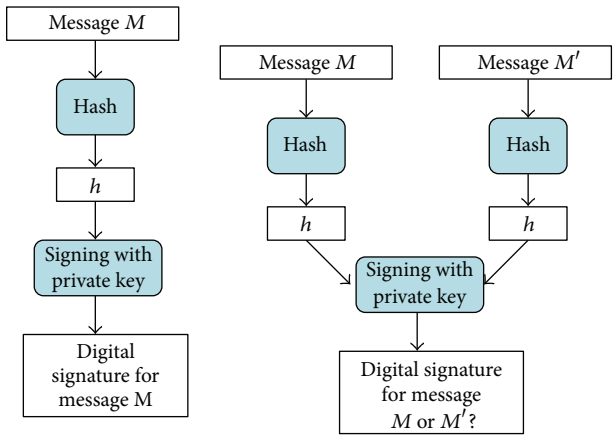

Figure 1: Digital signature process and collisions for hash function.

Table 1: Known attacks for cryptographic primitives.

Class	Algorithm	Known attack	Complexity
Block cipher	DES	Differential cryptanalysis	2^49
Block cipher	DES	Linear cryptanalysis	2^43
Stream cipher	RC4	Royal Holloway attack	2^24
Hash function	MD4	Differential cryptanalysis	2^8
Hash function	MD5	Differential cryptanalysis	2^19
Hash function	RIPEMD	Differential cryptanalysis	2^16

key encryption and MAC. Asymmetric key cryptography includes public key encryption and digital signature. Similar to cryptographic primitives, lifetime of cryptographic keys is shorter than preserved period. In this chapter, we discuss the lifetime of cryptographic keys and cryptographic outputs.

Symmetric keys are securely managed while the contents are preserved. For example, session keys for communications are used in just that session. But the encryption key for contents in storage should be managed for longer time. For long-term confidentiality, we should periodically update encrypted contents as the lifetime of cryptographic keys expires.

Asymmetric keys are usually included in X.509 certificate. There are two kinds of certificate even though the user adopts the same algorithm RSA for encryption and signature. Usually the valid period of the certificate is one year. For example, the valid period of certificates for internet banking is one year in republic of Korea. But the valid period of digital signature does not match that of the certificate. Since digital contents might be preserved in many decades, it is hard to guarantee the authenticity of the digital signature after decades. Moreover, we should manage private keys responding to certificates that have already expired because output contents live longer than certificates.

We also consider public security services dependent on cryptographic keys. Timestamp is a kind of digital signature applications. A single timestamp does not provide long-term security because it relies on a cryptographic hash function, a digital signature, or wide-visible media, which all are subject to security deterioration over time. We have to determine the lifespan of cryptographic key, signature, and archiving data very carefully. Table 2 shows the lifetime of cryptographic key and contents.

Table 2: Lifetime of cryptographic key and contents.

Class	Lifetime
Symmetric key	
Session key	Period of that session
Key for stored contents	Usually in years
Asymmetric key	
Certificate	1 or 3 years
Digital signature	Usually 5 years
Archives	Longer than decades

3.3. Other Issues for Long-Term Security. We need to consider other issues related to digital archiving systems such as the login system and the access control system. We examine not only the security of them but also how to delegate one's access right to others. Most login systems are based on cryptographic primitives. The password is stored in the system as encrypted form or its hash value. When underlying cryptographic primitives are broken, the administrator should change the whole login systems. Access control might be reinvestigated. Access right verification could be altered to be more sophisticated. For example, one rich man makes his will and stores it in the archive. After his demise, a bereaved family might access his will without revealing that to others.

We have to consider the content format too. There are continuously reports related to weakness of file format. We consider not only security but also backward compatibility for long-term availability. DRM (digital rights management) system is also very sensitive. Most DRM systems depend on system calls provided by the operating system. How long do we expect the backward compatibility of the operating system? We should consider a DRM system based on online cryptographic protocols.

Digital archives are designed for an outsourced storage service model which enables users to outsource the storage of their contents to remote storage service provider at a low cost. Since the management of storage is performed by a service provider, we should consider the balance between privacy and efficiency. For more privacy, we determine that the main actor for content protection is the user. But this may cause the security leakage by the individual's carelessness and inconsistency of the whole system security policy.

4. Requirements for Digital Archives

In this chapter, we discuss requirements for digital archive systems with issues in the previous chapter. For long-term confidentiality, there should be a process for updating encrypted data. The reencryption of large amounts of data must be done in a timely manner, especially if required for a key compromise or cryptographic primitives exploit. So we need robust high-speed encryption algorithm and reencryption procedure. For example, if a system chooses to

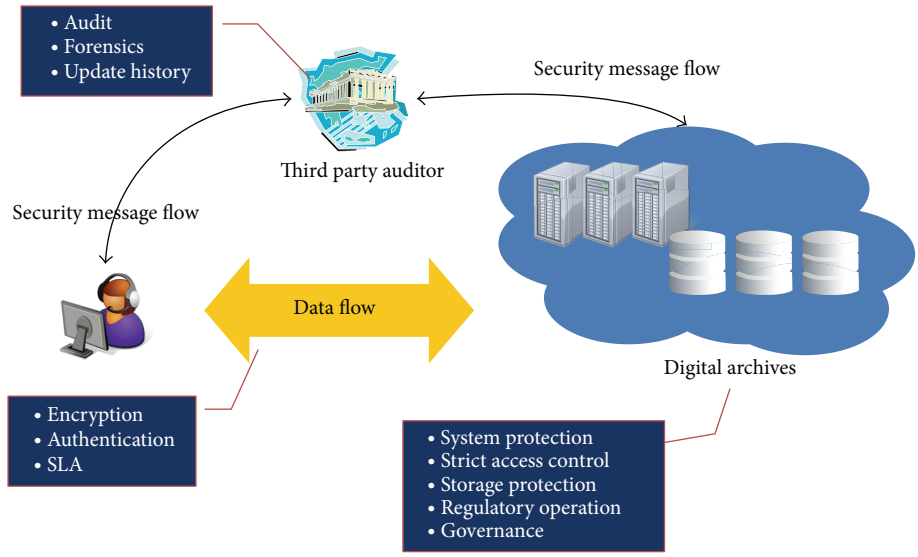

Figure 2: The third party auditor.

save time by encrypting over the old algorithm, it must have a way of dealing with key histories and key distribution. In contrast, if the system chooses to decrypt the data before applying the new algorithm, then it must have access to the user's key. In this case, the system should prevent a malicious user from the user's key and decrypted contents. For encryption update, we consider the following security requirements.

(i) Reencrypt encrypted data without revealing plain text to unauthorized users.

(ii) Make high-speed encryption for a large amount of data.

In Section 3, long-term authentication and integrity cannot be achieved by a single protection mechanism. Similar to confidentiality, we should consider a process for updating digital signatures. The digital signature update has to include some update history information. In that system, there are the following requirements.

(i) Make a signature update chain that cannot be reversed.

(ii) Secure delegation mechanism for signing keys.

We might consider the storage implementing with write-once media. This might be very expensive but easy for integrity. For more cost effective way, we can consider the following TTP service. We make another TTP called the third party auditor. Then digital archives stored sensitive log and audit in the third party auditor and the user verifies the log data with this TTP service. Figure 2 is a conceptual design of the third party audit.

For long-term security, we should carefully design an update procedure and period. There might be two kinds of update. One is a periodic update precautiously. The other is an emergency responding update for a key compromise or cryptographic primitives exploit. This emergency responding update is similar to disaster recovery. In this update procedure, the following should be included:

(i) policy for determining the update period;

(ii) monitoring plan;

(iii) update history management;

(iv) sealing process.

5. Conclusion

In this paper, we present some security issues related to digital archiving systems. Since digital archives preserve digital contents in very long term, we should consider the update of stored contents. These security issues should be considered before system design. And there should be the development cycle including design, implementation, monitoring and threat assessment, and periodic or emergency update. The focus of this paper was not to solve those problems which arise in long-term preservation but rather enumerate the requirements. We hope that by listing the security issues and requirements future efforts to build secure archives will be more focused. Moreover, we should consider that cryptographic approach cannot be a perfect solution without physical protection and well-established policy.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," *Proceedings of the 24th International Conference on Large Installation System Administration (LISA '10)*, 2010.

[2] G. Yamamoto, S. Oda, and K. Aoki, "Fast integrity for large data," *Proceedings of the Workshop on Software Performance Enhancement for Encryption and Decryption (SPEED '07)*, 2007.

[3] P. Maniatis, M. Roussopoulos, T. J. Giuli, D. S. H. Rosenthal, and M. Baker, "The LOCKSS peer-to-peer digital preservation system," *ACM Transactions on Computer Systems*, vol. 23, no. 1, pp. 2–50, 2005.

[4] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[5] J. Buchmann, A. May, and U. Vollmer, "Perspectives for cryptographic long-term security," *Communications of the ACM*, vol. 49, no. 9, pp. 50–55, 2006.

[6] J. Hughes and J. N. Roge, "Long-term security vulnerabilities of encrypted data," *Issues in Information Systems*, vol. 8, pp. 522–528, 2007.

[7] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton, Fla, USA, 1996.

[8] National Bureau of Standard, *Data Encryption Standard (DES)*, FIPS Publication 46, 1977.

[9] NIST, "Advanced Encryption Standard (AES)," FIPS Publication 197, 2001.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] R. Rivest and Network Working Group, "MD 4 message-digest algorithm," Tech. Rep. RFC 1320, 1992.

[12] Network Working Group and R. Rivest, "MD 5 Message-Digest Algorithm," RFC 1321, 1992.

[13] NIST, "Secure Hash Function," FIPS 180-1, 1995.

[14] NIST, "Digital Signature Standard (DSS)," FIPS 186, 1994.

[15] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 598–610, November 2007.

[16] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, "A secure cloud backup system with assured deletion and version control," in *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW '11)*, pp. 160–167, September 2011.

[17] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[18] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology—EUROCRYPT '93*, vol. 765 of *Lecture Notes in Computer Science*, pp. 386–397, Springer, Berlin, Germany, 1994.

[19] X. Wang and H. Yu, "How to break MD5 and other hash functions," in *Advances in Cryptology—EUROCRYPT 2005*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 19–35, Springer, Berlin, Germany, 2005.