LFSR-BASED CRYPTOGRAPHIC CHECKSUMS FOR SECURE BROADCASTING

Chung-Huang Yang
Basic Research Laboratory
Telecommunication Laboratories
Ministry of Transportation and Communications
P.O. Box 71
Chung-Li, TAIWAN 32099

楊 中皇
基本科技研究室
電信研究所
交通部
桃園縣楊梅鎮326
民族路三段551巷12號

E-mail: chyang%tl9000@tlrouter.motctl.gov.tw

ABSTRACT

In this paper, we present a scheme for generating cryptographic checksums to perform message authentication in a one-way broadcasting system. The proposed scheme is based on the use of clock-controlled LFSRs and is aimed at high-speed implementation for real-time applications.

INTRODUCTION

In a typical secure broadcasting system, source information (video, voice, images, text,..., etc.) are transmitted in a scrambled structure from the broadcast center and all the receivers get the same broadcasting signal consisting of scrambled information and access control information [1]. A set of randomly generated decryption keys, master keys, are stored in a physically secure and tamperproof module inside the legitimate receivers while the center keep a copy of corresponding encryption keys at a database against receiver's unique ID, etc. Successful descrambling occurs in authorized receivers when correctly encrypted descrambling keys were delivered from the broadcast center. Although the cryptographic algorithm selected for distributing descrambling key could either be a public-key algorithm or a private-key algorithm, DES-alike schemes are usually chosen and implemented in firmware or software to provide key distribution in a secure and cost-effective manner.

The receivers are individually addressable, meaning they can be directly controlled, using the unique master key, at any time from the broadcast center. Nevertheless, it is very inefficient to deliver the descrambling key to each receiver on an individual basis, therefore a key distribution hierarchy is normally developed. The idea is to form receiver groups and each member of the same group shares the identical distribution key [1]. In a three-level hierarchy of keys, the descrambling key is downloaded into receivers through a broadcasting message encrypted by the shared group key while the group keys is encrypted by the master key and distributed in advance into the secure module in the authorized receivers. Message for distributing the group key has to be extremely insensitive to transmission errors and a check must also be made by the receiver of the message that it has not been deliberately altered since it left the broadcast center. Furthermore, the delay and complexity that is required in implementing this message authentication must be satisfied with real-time and low-cost constrains in a commercial one-way broadcast system.

Cryptographic checksums (also known as message authentication codes, integrity check-values, modification detection codes, or message integrity codes) are used to detect unauthorized alternation of message being transmission between two mutually trusted parties [2-5]. Much like an ordinary checksum or cyclic redundancy check (CRC), a checksum is appended to the transmitted message and the number of checksum bits are generally less than the number of message bits to be transmitted. However, the cryptographic checksum will be dependent not only on the original message but also on a secret key known only between the sender and the intended receiver. When a message is transmitted, the broadcast center calculates a checksum using the agreed secret key and appends it to the message (Figure 1).

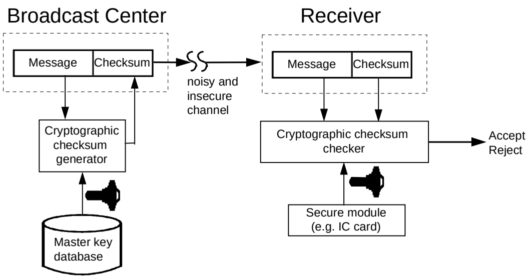
Broadcast Center
Receiver
Message
Checksum
noisy and insecure channel
Message
Checksum
Cryptographic checksum generator
Cryptographic checksum checker
Accept
Reject
Master key database
Secure module (e.g. IC card)

Figure 1. Message authentication using cryptographic checksum

At the other end, the receiver recomputes the checksum from the delivered message, using the same secret key stored in a tamperproof module, and compares it against the delivered checksum. The degree of security is dependent on the key length, strength of the cryptographic algorithm, and length of the checksum.

Several cryptographic checksum schemes have been proposed and analyzed, such as the one given on the international standard ISO 9797 [6], where checksum is implemented using the block chaining mode of a strong block cipher and the final encrypted block (or a part of it) is used as the checksum. Here we propose a new cryptographic checksum based on stream cipher cryptography. The main benefit of the proposed scheme is on its bit-serial high-speed computation capability while makes use of simple linear feedback shift-registers (LFSRs) [7] which is, in its own turn, a very popular module used in building complicated pseudorandom bit generators (PRBG) for video or audio scrambling. For example, the PRBGs proposed by the European Broadcast Unit [8] and the Japan's Ministry of Posts and Telecommunications [9] are both based on the idea of output-control LFSRs and the video signal is manipulated and scrambled under a PRBG controlled by a secret key.

A LFSR is consisted of a shift register of n flip-flops and a feedback connection such that each element of the output sequence is a fixed linear function of the previous n elements. The feedback connections will decide the period and statistical behavior of the output sequence. By properly selecting the feedback connection, period of an n-element LFSR output sequence will be 2^n-1 for any non-zero initial state. But, due to its linearity, such scheme is unsuitable to combat deliberate modification of transmitted message, assuming that the initial state is chosen as the secret key, final state is the checksum, and feedback function is added (XORed) with the message in a bit-serial form.

THE PROPOSED SCHEME

To strength LFSR-based checksum, we might use nonlinear feedback shift-registers [7] which could produce de Bruijn sequences of period 2^n; but at present such circuits are difficult to implement efficiently. Instead, we will follow the sequence generators proposed in [10] on the basis of combining the work of two maximum-length LFSRs by mutual stop-and-go clock control.

The proposed checksum scheme is shown in Fig. 2. In the scheme, we have a pair of n-stage maximal-length LFSRs, at each clock one of the feedback functions is added (XORed) with the message in bit-serial form, and the contents of both LFSRs at end is used as the checksum. The LFSRs are preloaded with the secret key and other parameters (such as message identifier, message length, time stamp, random number, and/or message sequence number) for security purposes .

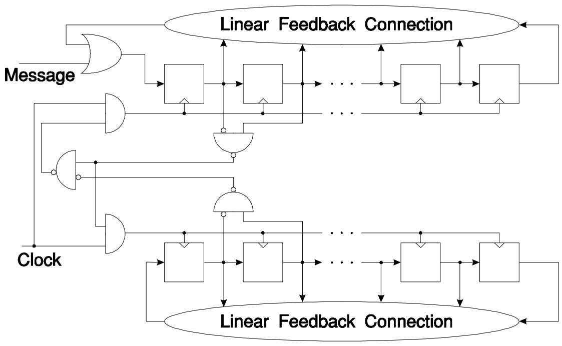

Figure 2. Circuit diagram of the proposed cryptographic checksum

The state diagram for autonomous behavior (message input is ignored) of the scheme, as illustrated in Fig. 3, has been theoretically analyzed [10] and is consisted of a total of $3 \times 2^{n-2}-1$ cycles with identical cycle length of $5 \times 2^{n-2}-1$. The number of stages in both LFSRs, n, is chosen as $n = 34$, or 50, or 56, or 74,..., etc., such that the cycle length $5 \times 2^{n-2}-1$ is a prime number. The state diagram will have the following distinct characteristics: (1). No state can have more than two predecessors; (2). Every branch to a cycle starts with a source state, state which does not have predecessor, has length of one, and at any non-source state there is at most one entering branch.

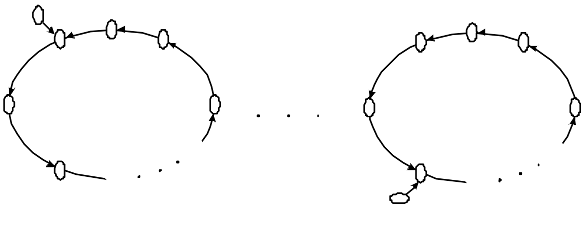

Figure 3. State diagram for autonomous behavior of the proposed scheme

With regarding the security, no general criterion at the present time has been developed to certify the security of cryptographic checksum schemes, and many scheme have been proposed and then broken (i.e., found more than one message which gives rise to a particular checksum). Any new scheme ought to show its merits in comparing with existing schemes. Here the proposed scheme has advantages of using existing circuits which might also be utilized for scrambling purpose, providing a guaranteed key-independent large period at autonomous mode, achieving in high speed (clock rate is limited by a few gate delay), and allowing bit-serial input of message which is the typical case in telecommunication.

CONCLUSIONS

Cryptographic checksum is used as a data integrity mechanism to detect that message has not been altered in an unauthorized manner. A checksum is appended to the transmitted message, using a secret key, at the sender and when the message is modified unintentionally or maliciously, such changes have a high probability of being detected after computation of the new checksum at the receiver end. We have proposed in this brief paper a bit-serial cryptographic checksum scheme, based on clock-controlled LFSRs which produces state cycles with a prime period in autonomous behavior. The scheme was designed to be also used as a pseudorandom bit generators for video or audio scrambling.

REFERENCES

1. CCIR Report 1079-1, *General Characteristics of a Conditional-Access Broadcasting System*, 1986.

2. R.R. Jueneman, S.M. Matyas, and C.H. Meyer, "Message Authentication with Manipulation Detection Codes," *IEEE Symp. on Security and Privacy*, 1983, pp. 33-54.

3. C. Mueller-Schloer, "DES-generated Checksums for Electronic Signatures," *Cryptologia*, Vol. 7, No. 3, July 1983, pp. 257-273.

4. F. Cohen, "A Cryptographic Checksum for Integrity Protection," *Computers and Security*, Vol. 6, 1987, pp. 505-510.

5. H. Beker, "The User of Automated Cryptographic Check-sums," *IFIP Computer Security in the Age of Information*, 1989, pp. 75-81.

6. ISO/IEC 9797:1989, *Data cryptographic techniques - Data integrity mechanism using a cryptographic check function employing a block cipher algorithm*.

7. W. Golomb, *Shift Register Sequence*, Aegean Park Press, 1982.

8. European Broadcast Unit (EBU), *Specification of the Systems of the MAC/Packet Family*, EBU Tech. 3258, 1986.

9. Official Gazette, Japan Ministry of Posts and Telecommunications, January 25, 1990, No. 36 (in Japanese) (日本官報1990年1月25日郵政省告示第36號).

10. K.C. Zeng, C.H. Yang, and T.R.N. Rao, "Large Primes in Stream Cipher Cryptography," *Advances in Cryptology - Auscrypt'90*, Springer-Verlag Lecture Notes in Computer Science Vol. 453, 1990, pp. 194-205.