CDMA Physical Layer Built-in Security Enhancement

Jian Ren Tongtong Li
2120 Engineering Building
Department of Electrical & Computer Engineering
Michigan State University
East Landing, MI 48864-1226
Email: {renjian, tongli}@egr.msu.edu

Abstract— **Served as one of the most widely used wireless airlink interface, CDMA has been identified as a major technique for 3G wireless communications. The current IS-95 CDMA provides a near-satisfactory security solution to voice centric wireless communications, since generally each voice conversation only lasts a very short period of time. However, the security features provided by IS-95 CDMA are far from adequate and being acceptable when used for data communications. In this paper, the security weakness of the existing CDMA airlink interface is analyzed. Encrypted key stream based on advanced encryption standard (AES) is proposed to be used in the scrambling process, instead of using the scrambling sequence generated from the 42-bit linear feedback shift register (LFSR) as in IS-95. Ensured by AES, physical layer built-in security of the proposed scheme is much stronger than that of the IS-95 system.**

I. Introduction

As people are relying more and more on wireless communication networks for critical information transmission, security has become an urgent issue and a bottleneck for new wireless communication services such as wireless mobile Internet and e-commerce [7]. For military communications where information transmission heavily relies on wireless networks (for example, from aircraft to aircraft, from aircraft to ground control center etc.), security and reliability of the wireless communication systems is of number one priority, especially in national defense and emergency response to abrupt enemy attacks.

It is fairly common to authenticate users and make the users be responsible for their actions in data networks. Generally, a logic and also a physical boundary exist for data networks, where access control and comprehensive security services can be implemented. However, wireless network is an untrustable environment. Unknown users are difficult to identify and hold accountable for damages, user authentication methods are therefore required to limit the amount of damage and to ensure that the users take responsibility for their actions and can be tracked for security concerns. An authenticated user can gain access to services and the commodity Internet from which unauthenticated users are blocked. This is especially important for national security.

Another factor that contributes to the insecurity of wireless communications is that the current wireless communication techniques were initially designed mainly for voice communications, instead of data communications which require much stronger security. The existing techniques such as spread spectrum and long-code mask are primarily designed for communication performance and reliability, though they can indeed provide very limited security. The fast computational speed improvement, rapid receiver technology advance and price declination facilitate the malicious attackers with an easy access to the wireless communication channels in the air. The security techniques that are based on the possession of wireless receivers and limited computations are out-of-date and can not provide adequate security protections. The ultimate approach to secure wireless communications has to rely on modern cryptography, such as pseudo-random sequences design, data encryption and access control.

This paper aims to improve the physical layer built-in security of wireless communication systems by combining cryptographic techniques with modulation techniques and multiple access techniques in transmitter and receiver design, as an effort to design more secure and reliable wireless communication systems.

In the current commercial CDMA systems, each user's signal is first spread using a code sequence (known as *channelization code*) spanning over just one symbol or multiple symbols. The spread signal is then further scrambled using

0-7803-7954-3/03/$17.00 ©2003 IEEE.

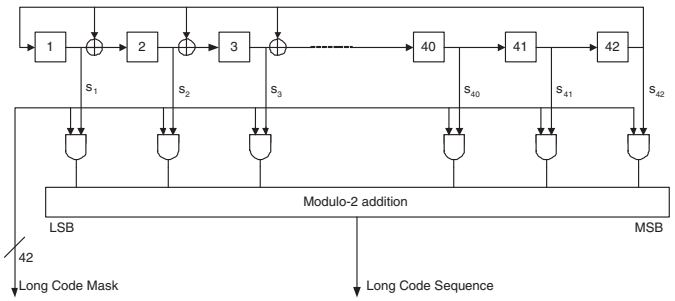

Fig. 1. IS-95 long-code Generator

a pseudo-random sequence, to randomize the interference and meanwhile make it difficult to intercept and detect the transmitted signal. To recover the desired user's signal, one has to know both the user's channelization code and scrambling code. This is known as the built-in security feature of the CDMA systems. However, the system is fragile to hostile security attacks.

More specifically, the security of CDMA system mainly relies on the *long-code generator* consists of a 42-bit *long-code mask* generated by a 42-bit linear feedback shift registers (LFSRs). The maximum complexity to recover the 42-bit long-code mask is $O(2^{42})$. However, if an eavesdropper can obtain 42 bits of plaintext-ciphertext pairs, then the long-code mask can be recovered after eavesdropping the transmission on the traffic channel for about one second [16].

In this paper, we propose to enhance the built-in security of CDMA systems by applying cryptographic algorithm to the scrambling process.

II. Security of IS-95 CDMA

In IS-95 CDMA systems, *long-code mask sequence* [2], [10] is used for signal scrambling and to provide voice privacy in the physical layer. The long-code sequence is generated by the linear feedback shift register (LFSR) as shown in Figure 1.

The long-code generator consists of a shared 42-bit number called *long-code mask* and a 42-bit LFSR specified by the following characteristic polynomial:

$$\begin{aligned} & x^{42}+x^{35}+x^{33}+x^{31}+x^{27}+x^{26}+x^{25} \\ & +x^{22}+x^{21}+x^{19}+x^{18}+x^{17}+x^{16} \\ & +x^{10}+x^{7}+x^{6}+x^{5}+x^{3}+x^{2}+x+1. \end{aligned} \tag{1}$$

Since the long-code mask has only 42-bit stage functions as user keys, the maximum complexity to recover the 42-bit long-code mask is $O(2^{42})$. However, if an eavesdropper can obtain 42 bits of plaintext-ciphertext pairs, then the long-code mask can be recovered after eavesdropping the transmission on the traffic channel for about one second.

In fact, although different base stations use different long-code masks, the long-code sequences are all essentially generated by the same LFSR in equation (1). Let $M = [m_1, m_2, \cdots, m_{42}]$ denote the 42-bit mask for a base station and $S(t) = [s_1(t), s_2(t), \cdots, s_{42}(t)]$ denote the state of the LFSR at time instance t. The long-code sequence $c(t)$ at time t can thus be represented as

$$c(t) = m_1 s_1(t) + m_2 s_2(t) + \cdots + m_{42} s_{42}(t), \tag{2}$$

where the additions are modulo-2 additions.

Since $s_1(t), s_2(t), \cdots, s_{42}(t)$ are the outputs of the same LFSR, they should all be the same except for a phase difference, i.e.,

$$s_{42}(t) = s_{41}(t-1) = \cdots = s_1(t-41).$$

Therefore, according to equation (1), for $a = [a_1, a_2, \cdots, a_{42}]$ consisting of the coefficients of equation (1), we have

$$\begin{aligned} s_i(t) &= a_1 s_{i-1}(t) + a_2 s_{i-2}(t) + \cdots + a_{42} s_{i-42}(t) \\ &= a_1 s_i(t-1) + a_2 s_i(t-2) + \cdots \\ &\quad + a_{42} s_i(t-42). \end{aligned} \tag{3}$$

Substitute (3) into (2), we have

$$\begin{aligned} c(t) &= \sum_{i=1}^{42} m_i s_i(t) \\ &= \sum_{i=1}^{42} m_i \Big(\sum_{j=1}^{42} a_j s_i(t-j) \Big) \\ &= \sum_{j=1}^{42} a_j \Big(\sum_{i=1}^{42} m_i s_i(t-j) \Big) \\ &= \sum_{j=1}^{42} a_j c(t-j) \end{aligned}$$

Define

$$A = \begin{bmatrix} a_1 & 1 & 0 & \cdots & 0 \\ a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{41} & 0 & 0 & \cdots & 1 \\ a_{42} & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{4}$$

then it follows that

$$\begin{aligned} & [c(t), c(t-1), \cdots, c(t-41)] \\ = \; & [c(t-1), c(t-2), \cdots, c(t-42)] * A. \end{aligned} \tag{5}$$

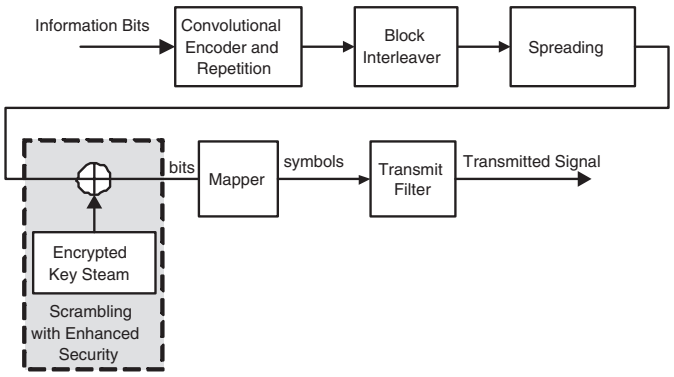

Fig. 2. Proposed CDMA Physical Layer Security Enhancement

Let $C(t) = [c(t), c(t-1), \cdots, c(t-41)]$, then for any $n \geq t$, from equation (5) we have

$$C(n) = C(t) * A^{n-t}. \tag{6}$$

In equation (6), A is defined based on the linear feedback shift register sequence defined in equation (1). Therefore, as long as $C(t)$ for a time instance t is known, then the subsequent sequences will be known. In other words, as long as an eavesdropper can intercept/recover up to 42 continuous long-code sequence bits, then the whole long-code sequence can be generated. Therefore, the long-code sequence is vulnerable under ciphertext-only attacks. In [16], it was demonstrated that the long-code sequence for CDMA downlink traffic channel can be recovered in one frame. That is, it is not necessary to try the 2^{42} initial stage in order to recover the long-code sequence and the complexity is much lower.

III. Security Enhancement of the Scrambling Process Based on AES

As we mentioned in Section II, in IS-95 system, user privacy is provided by scrambling the chip-rate spread signal using a long-code pseudo-noise sequence, which is generated by a 42-bit LFSR. In which, the base station and the mobile share the 42-bit initial state of the LFSR as the secret key, which is also used for synchronization. To enhance the physical layer built-in security of CDMA systems, in this paper, we propose to generate the scrambling sequence using the advanced encryption standard (AES as known as Rijndael).

Rijndael was identified as the new Advanced Encryption Standard (AES) in October 2, 2000. Rijndael's combination of security, performance, efficiency, ease of implementation and flexibility make it an appropriate selection for the AES. It is also a very good performer in both hardware and software across a wide range of computing environments regardless of its use in feedback or non-feedback modes. Its key setup time is excellent, and its key agility is good. Rijndael's very low memory requirements make it very well suited for restricted-space environments such as mobile handset to achieve excellent performance.

Additionally, Rijndael is designed with some flexibility in terms of block and key sizes, and the algorithm can accommodate alterations in the number of rounds. Rijndael's internal round structure has good potential for parallelism.

The proposed secure scramble process has three steps:

1) The base station and the mobile share a common initial vector and a common secret key;
2) The secure scrambling sequence is generated using the initial vector and the secret key through AES operations;
3) The scrambling process is realized by applying the secure scrambling sequence to the chip-rate spread signal.

The basic unit for processing in the AES algorithm is a byte, or a sequence of 8 bits treated as a single entity. The input, output and Cipher Key bit sequences are processed as arrays of bytes that are formed by dividing these sequences into groups of 8 contiguous bits to form arrays of bytes. The AES enables three different key sizes be used in encryption and decryption: 128 bits, 192 bits and 256 bits. It also has three allowable block sizes: 128 bits, 192 bits and 256 bits.

The scrambling sequence can be generated from the initial vector and the secret key in either output feedback mode (OFB) or cipher feedback mode (CFB). In the following, we briefly describe the encryption process for the scrambling sequence generation. For simplicity, we limit the block size and the key size to 128 bits in this paper and demonstrated the encryption process using OFB mode, in which the scrambling sequence is produced by repeatedly encrypting the 128-bit (or 16 bytes) initialization vector, denoted by V.

At the start of the cipher, V is copied to a 4×4 State array. Suppose the input byte string is $V_0, V_1, \cdots, V_{15}$, then the following array, called *State Array* will be generated:

$$\begin{array}{|c|c|c|c|} \hline V_0 & V_4 & V_8 & V_{12} \\ \hline V_1 & V_5 & V_9 & V_{13} \\ \hline V_2 & V_6 & V_{10} & V_{14} \\ \hline V_3 & V_7 & V_{11} & V_{15} \\ \hline \end{array} \triangleq \begin{array}{|c|c|c|c|} \hline s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ \hline s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ \hline s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ \hline s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \\ \hline \end{array}$$

where each $s_{i,j} = V_{i+4j}$ is a byte, for $i, j = 0, 1, 2, 3$.

Four different transformations are defined to process the State: SubBytes(), ShiftRows(), MixColumns(), and AddRoundKey(). In SubBytes() transformation, a non-linear bytes substitution operates on each byte of the State array in-

dependently using a substitution table (S-Box), which requires only a 256 entries. In fact, the S-box can also be generated mathematically through two different mathematics operations:

1) Map each byte in the State array to its multiplicative inverse in the finite field $GF(2^8)$; the value 00 is mapped to itself.
2) Denote each byte of the inverse State entry as (b_3, b_2, b_1, b_0) and apply the following transformation to each bit of each byte in the State array:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

In the ShiftRows() transformation, the bytes in the last three rows of the State are cyclically shifted left by 1, 2, and 3 positions respectively. Now the State Array is becoming

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,2}$	$s_{1,3}$	$s_{1,0}$	$s_{1,1}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,2}$	$s_{3,3}$	$s_{3,0}$	$s_{3,1}$

The MixColumns() transformation operates on the State column-by-column, where each column is treated as a four-term polynomial. The columns are defined over $GF(2^4)$. The columns are multiplied modulo x^4+1 with a fixed polynomial $a(x)$

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}.$$

That is

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$\Longrightarrow$

$s'_{0,0}$	$s'_{0,1}$	$s'_{0,2}$	$s'_{0,3}$
$s'_{1,0}$	$s'_{1,1}$	$s'_{1,2}$	$s'_{1,3}$
$s'_{2,0}$	$s'_{2,1}$	$s'_{2,2}$	$s'_{2,3}$
$s'_{3,0}$	$s'_{3,1}$	$s'_{3,2}$	$s'_{3,3}$

where

$$\begin{bmatrix} s'_{0,i} \\ s'_{1,i} \\ s'_{2,i} \\ s'_{3,i} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,i} \\ s_{1,i} \\ s_{2,i} \\ s_{3,i} \end{bmatrix}, \text{ for } 0 \le i \le 4.$$

The final transformation is the AddRoundKey() transformation. In this step, a Round Key is added to the State by a simple bitwise XOR operation. Each Round Key consists of 4 bytes from the key schedule. Those 4 words are each added into the columns of the State, such that

$$[s'_{0,i}, s'_{1,i}, s'_{2,i}, s'_{3,i}] = [s_{0,i}, s_{1,i}, s_{2,i}, s_{3,i}] \oplus [w_{round*4+i}]$$

where $0 \le i < 4$. $[w_i]$ are the key schedule. The *round* is a value in the range $0 \le round \le N_r$ (the number of rounds) described hereafter.

The AES algorithm takes the Cipher Key, K, and performs a Key Expansion routine to generate the round keys. The Key Expansion generates a total of $N_b(N_r+1)$ bytes arranged as an $N_b \times (N_r+1)$ array: N_b bytes for initial and N_b for each of the N_r rounds. The resulting key schedule consists of a linear array of 4-byte words, denoted $[w_i]$, where $0 \le i < N_b(N_r+1)$.

Denote the columns of K as $K_i, i = 0, 1, 2, 3$. Now we will expand K by adjoining 40 more columns as follows: Suppose columns up through K_{i-1} have been defined.

$$K_i \triangleq \begin{cases} K_{i-4} \oplus K_{i-1}, & \text{if } i \not\equiv 0 \ (\mod 4) \\ K_{i-4} \oplus T(K_{i-1}), & \text{if } i \equiv 0 \ (\mod 4) \end{cases}$$

where $T(K_{i-1})$ is a transformation of K_{i-1} through the following four steps:

1) Suppose $K_{i-1} = (a,\ b,\ c,\ d)^t$. We first shift K_{i-1} to $(b,\ c,\ d,\ a)^t$, where t denotes the transpose.
2) Replace each of these bytes with the corresponding element in the S-box to get 4 bytes e, f, g, h.
3) Compute the round constant $r_i = 00000010^{(i-4)/4}$ in $GF(2^8)$.
4) $T(K_{i-1}) = (e \oplus r_i,\ f,\ g,\ h)^t$.

The **round key** for the ith round consists of:

$$K_{4i},\ K_{4i+1},\ K_{4i+2},\ K_{4i+3}.$$

At this stage, the first 128 bits of the secure scrambling sequence is obtained. This 128-bit segment of scrambling sequence obtain in the first step is then used as the initial vector of the above process to obtain the second 128-bit segment of the scrambling sequence. In other words, the output of each round is used as the input for the next round of AES operations, and the process continuous if necessary.

IV. Security of the Proposed Scrambling Process

In this section, we use Data Encryption Standard (DES) as a benchmark to calculate the number of possible keys of AES and that of IS-95 sequence. The number of keys determines the effort required to crack the cryptosystem by trying all possible keys.

The most important reason for DES to be replaced by AES is that it is becoming possible to crack DES by trying all possible keys. Single DES uses 56 bits encryption key, which means there are approximately 7.2×10^{16} possible DES keys. In the late 1990s, specialized "DES Cracker" machines were

built that could recover a DES key after a few hours. In other words, by trying possible key values, the hardware could determine which key was used to encrypt a message [1]. In comparing with DES, IS-95 has only 42-bit shared secret. The approximate number of keys is only about 4.40×10^{12}, which is only about 10^{-4} of that of the number of DES 56-bit keys. This makes is possible to break the IS-95 long-code mask almost in real time.

On the other hand, AES specifies three key sizes: 128, 192 and 256 bits. In decimal terms, this means that there are approximately:

- 3.4×10^{38} possible 128-bit keys;
- 6.2×10^{57} possible 192-bit keys;
- 1.1×10^{77} possible 256-bit keys.

Thus, there are on the order of 10^{21} times more AES 128-bit keys than DES 56-bit keys. Assuming that one could build a machine that could recover a DES key in a second (i.e., try 2^{55} keys per second), as we can see, this is a very ambitious assumption and far from what we can do today, then it would take that machine approximately 149 thousand-billion (149 trillion) years to crack a 128-bit AES key. To put that into perspective, the universe is believed to be less than 20 billion years old.

V. Key Exchange

Wireless traffic is transmitted in the air where anyone with the technology, including the malicious users, can intercept it. As part of data confidentiality service, a proper key exchange mechanism is crucial to the security of the system. In fact, key exchange is often closely related to authentication, which is another major security service for data communications.

A mutual authentication between the mobile handset and base station (BS) or mobile switching center (MSC) is the most practical and dominant technique to eliminate unauthorized access. Key exchange should follow a successful authentication between the mobile handset and the BS or MSC.

The nature of mobility and the diversity of wireless communications make X.509 digital certificates a prominent solution for authentication and key exchange. There are two feasible implementations for X.509 digital certificates. One approach is to implement all the cryptographic services in the physical layer though a dedicated processor/chip. The advantage of this approach is to apply the best security services without interfering the existing airlink standard. The other approach is to implement the service in the network layer. In either case, limited network management is required in order to acquire digital certificates from a public certificate authority, which is a similar process as a phone number is obtained from a yellow page.

VI. Conclusion

In this paper, security weakness of IS-95 CDMA system is analyzed and an encryption based secure scrambling process is presented. Instead of using the long-code sequence generated by a 42-bit LFSR as in IS-95, the scrambling sequence is generated through AES operations. Therefore, the security of the proposed scheme is ensured by that AES, and the physical layer built-in security of the CDMA system is significantly increased with limited complexity load. Options of key exchange and authentication are also provided in this paper.

References

[1] EFF DES Cracker Project. Cracking DES. `http://www.eff.org/descracker/`.

[2] V.k. Gray. *IS-95 CDMA and cdma2000*. Prentice Hall, 2000.

[3] M. Honig, U. Madhow, and S. Verdù. Blind adaptive multiuser detection. *IEEE Trans. on Information Theory*, IT-41, July 1995.

[4] Joan Daemen and Vincent Rijmen. AES Proposal: Rijndael, March 1999.

[5] Tongtong Li and J. K. Tugnait. Super-exponential methods for blind detection of asynchronous CDMA signals over multipath channels. In *Proceeding of ICC 2001*, Helsinki, Finland, June 11-14 2001.

[6] National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard (AES), November 2001. `http://csrc.nist.gov/CryptoToolkit/aes/`.

[7] R.K. Nichols and P. C. Lekkas. *Wireless Security: Models, Threats, and Solutions*. McGraw-Hill Telecom, 2002.

[8] J.G. Proakis. *Digital Communications*. McGraw-Hill, 3nd edition, 1995.

[9] Theodore S. Rappaport. *Wireless Communications – Principles and Practices*. Prentice Hall, second edition, 2002.

[10] TIA/EIA/IS-95-B. *Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System*, 1998.

[11] M. Torlak and G. Xu. Blind multiuser channel estimation in asynchronous CDMA systems. *IEEE Trans. on Signal Processing*, SP-45:137–147, January 1997.

[12] J. K. Tugnait and Tongtong Li. Blind asynchronous multiuser CDMA receivers for ISI channels using code-aided CMA. *IEEE Journal on Selected Areas in Communications*, JSAC-19, August 2001.

[13] J. K. Tugnait and Tongtong Li. Blind detection of asynchronous CDMA signals in multipath channels using code-constrained inverse filter criteria. *IEEE Trans. on Signal Processing*, SP-49, July 2001.

[14] S. Verdú. Multiuser detection. *Advances in Statistical Signal Processing*, 2:369–409, May 1993.

[15] X. Wang and H.V. Poor. Blind adaptive multiuser detection in multipath CDMA channels based on subspace tracking. *IEEE Trans. on Signal Processing*, SP-46:3030–3043, November 1998.

[16] Muxiang Zhang, Christopher Carroll, and Agnes Hui Chan. Analysis of IS-95 CDMA voice privacy. In *Selected Areas in Cryptography*, pages 1–13, 2000.