Physical Layer Built-in Security Analysis and Enhancement of CDMA Systems

Tongtong Li Jian Ren Qi Ling
Department of Electrical & Computer Engineering,
Michigan State University, East Lansing, Michigan 48824, USA.
Email: {tongli,renjian,lingqi}@egr.msu.edu

Abstract— **Historically developed for secure communication and military use, CDMA is now serving as one of the most widely used wireless airlink interface and has been identified as a major technique for 3G wireless communications. In addition to the wide bandwidth and low power spectrum density which make CDMA signals robust to narrow band jamming and easy to be concealed within the noise floor, the physical layer built-in information privacy of CDMA system is provided by pseudo-random scrambling. In this paper, first, the physical layer security weakness of the operational IS-95 CDMA airlink interface is analyzed. Secondly, based on the advanced encryption standard (AES), we propose to enhance the physical layer built-in security of CDMA systems through secure scrambling. Performance analysis demonstrates that while providing significantly improved information privacy, CDMA system with secure scrambling has comparable computational complexity and system performance with that of the IS-95 system. Moreover, it is shown that by scrambling the training sequence and the message sequence separately with two independent scrambling sequences, both information privacy and system performance can be further improved. The proposed scheme can readily be applied to 3G systems and IEEE 802.11b WLAN systems.**

I. Introduction

In CDMA systems, each user is assigned a specific spreading sequence to modulate its message signal. The spreading process increases the bandwidth of the message signal by a factor N, known as spreading factor or the processing gain, and meanwhile reduces the power spectrum density of the signal also by a factor N. With large bandwidth and low power spectrum density, CDMA signals are resistant to malicious narrow band jamming and can easily be concealed within the noise floor, preventing from being detected by an unauthorized person. Moreover, the message signal can not be recovered unless the spreading sequence is known, makes it difficult for an unauthorized person to intercept the signal. Due to high spectrum efficiency and simplicity in system planning, CDMA is used in the US digital cellular standard IS-95 and has been identified as the major modulation technique for third generation (3G) wireless communications.

Relied on the long pseudo-random spreading sequence generator, the operational CDMA system (IS-95) can provide a near-satisfactory physical layer built-in 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 these systems 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 long code mask and 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 significantly improved compared to that of the IS-95 system. The proposed scheme can readily be applied to 3G systems and IEEE 802.11b WLAN systems, in combination with MAC layer and network layer security protocols, wireless network security is ensured from both the physical layer and upper layers.

II. Physical Layer Security Evaluation of the Operational IS-95 CDMA System

In the operational direct sequence CDMA (DS-CDMA) systems, as shown in Figure 1, 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 a pseudo-random sequence, to randomize the interference and meanwhile make it difficult to intercept and detect the transmitted signal. It is impossible to recover the desired user's signal without knowing both the user's channelization code and scrambling code. This is known as the built-in security feature of the CDMA systems.

Since the channelization codes are chosen to be Walsh codes, which are easy to generate, the physical layer

This paper is supported in part by MSU IRGP.

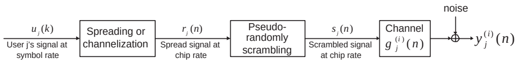
$u_j(k)$
User j's signal at symbol rate
Spreading or channelization
$r_j(n)$
Spread signal at chip rate
Pseudo-randomly scrambling
$s_j(n)$
Scrambled signal at chip rate
Channel $g_j^{(i)}(n)$
noise
$y_j^{(i)}(n)$

Fig. 1. Block diagram of a long code DS-CDMA System

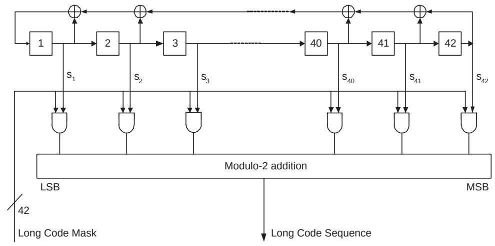

Fig. 2. IS-95 long code Generator

built-in security of CDMA systems mainly relies on the long pseudo-random scrambling sequence, also known as long code. In IS-95, the long code generator consists of a 42-bit number called *long code mask* and a 42-bit linear feedback shift register (LFSR) specified by the following characteristic polynomial:

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

where the 42-bit long code mask is shared between the mobile and the base station. As shown in Figure II, each chip of the long code is generated by the modulo-2 inner product of a 42 bit mask and the 42 bit state vector of the LFSR.

Let $M=[m_1,m_2,\cdots,m_{42}]$ denote the 42-bit mask 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

$$c(t)=m_1s_1(t)+m_2s_2(t)+\cdots+m_{42}s_{42}(t). \quad (2)$$

where the additions are modulo-2 additions.

As is well known, for a sequence generated from an n-stage linear feedback shift register, if an eavesdropper can intercept a $2n$-bit sequence segment, then the characteristic polynomial and the entire sequence can be reconstructed according to the Berlekamp-Massey algorithm [6]. This leaves an impression that the maximum complexity to recover the long code sequence $c(t)$ is $O(2^{84})$. However, for IS-95, since the characteristic polynomial is known to the public, an eavesdropper only needs to obtain 42 bits of the long code sequence to determine the entire sequence [14]. That is, the maximum complexity to recover the long code sequence $c(t)$ is only $O(2^{42})$

In fact, 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.,

$$s_{42}(t)=s_{41}(t-1)=\cdots=s_1(t-41). \quad (3)$$

Let $a=[a_1,a_2,\cdots,a_{42}]$ denote of the coefficient vector of the characteristic polynomial in equation (1), then it follows from (3) that

$$\begin{aligned}s_i(t)&=a_1s_{i-1}(t)+a_2s_{i-2}(t)+\cdots+a_{42}s_{i-42}(t)\\&=a_1s_i(t-1)+a_2s_i(t-2)+\cdots+a_{42}s_i(t-42)\end{aligned} \quad (4)$$

Substitute (4) into (2), we have

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

Define

$$A=\begin{bmatrix}a_1&1&0&\cdots&0\\a_2&0&1&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\a_{41}&0&0&\cdots&1\\a_{42}&0&0&\cdots&0\end{bmatrix}, \quad (5)$$

then it follows that

$$\begin{aligned}&[c(t),c(t-1),\cdots,c(t-41)]\\=\;&[c(t-1),c(t-2),\cdots,c(t-42)]*A.\end{aligned} \quad (6)$$

Let $C(t)=[c(t),c(t-1),\cdots,c(t-41)]$, then for any $n\geq t$, from equation (6) we have

$$C(n)=C(t)*A^{n-t}. \quad (7)$$

Therefore, as long as $C(t)$ for a time instance t is known, then the entire sequence can be recovered. 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 regenerated. Therefore, the long code sequence is vulnerable under ciphertext-only attacks.

Once the long code sequence is recovered, then the desired user's signal can be recovered through signal separation and extraction techniques. If the training sequence is known, simple receivers, for example, the Rake receiver, can be used to extract the desired user's signal. Even if the training sequence is unknown, desired user's signal can still be recovered through blind multiuser detection and signal separation algorithms, such as [1], [3], [12], [13].

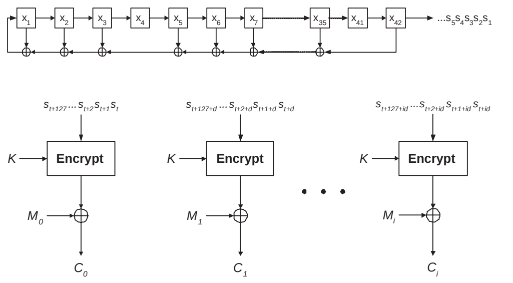
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_{35}
x_{41}
x_{42}
$\ldots s_5 s_4 s_3 s_2 s_1$
$s_{t+127} \cdots s_{t+2} s_{t+1} s_t$
$s_{t+127+d} \cdots s_{t+2+d} s_{t+1+d} s_{t+d}$
$s_{t+127+id} \cdots s_{t+2+id} s_{t+1+id} s_{t+id}$
K
Encrypt
K
Encrypt
K
Encrypt
M_0
M_1
M_i
C_0
C_1
C_i

Fig. 3. Proposed CDMA Physical Layer Secure Scrambling

III. Security Enhancement of the Scrambling Process Based on AES

As can be seen from the previous sections, the physical layer security of CDMA systems relies on the scrambling process, and the built-in information privacy provided by the operational IS-95 system is far from adequate. In this paper, to enhance the physical layer built-in security of CDMA systems, we propose to generate the scrambling sequence using the advanced encryption standard (AES), also known as Rijndael.

Rijndael was identified as the new AES in October 2, 2000. Rijndael's combination of security, performance, efficiency, ease of implementation and flexibility makes it an appropriate selection for the AES. Rijndael is a good performer in both hardware and software across a wide range of computing environments. Its low memory requirements make it very well suited for restricted-space environments such as mobile handset to achieve excellent performance. A brief introduction of AES can be found in the Appendix of this paper, and please refer to [5] for more details.

The proposed secure scrambling scheme aims to increase the physical layer built-in security of CDMA systems, prevent exhaustive key search attack, while minimizing the changes required to the operational IS-95 standard. As shown in Figure III, the proposed secure scrambling is essentially a counter mode AES. In Figure III, $s_0 s_1 s_2 \cdots$ represents the output of the LFSR characterized by (1) as in the IS-95 system, K is the 128 bits common secret encryption key shared between the base station and the mobile station (K can also be 192 bits or 256 bits, as specified in the AES algorithm), and $M_0, M_1, \cdots, M_i$ denote succesive message blocks with the same size as K, d is the shift between the successive inputs to the AES engine. If the input to the i-th encryption block is $s_{t+id}, s_{t+1+id}, \cdots, s_{t+127+id}$ with initial delay t, then the input to the $i+1$-th block is $s_{t+(i+1)d}, s_{t+1+(i+1)d}, \cdots, s_{t+127+(i+1)d}$,. The selection of d should maximize the diversity between different inputs to the AES engine, which can be achieved by requiring d and $2^{42}-1$ be relatively prime. In other words, d should not be divided by 3, 7, 43 and 127.

The secure scrambling process can be summarized as:

1) The base station and the mobile station share a common initial state for the LFSR and an L-bit (L=128, 192 or 256) common secret encryption key K;
2) The long scrambling sequence is generated through encryption of a particular segment of the sequence generated from the LFSR using the shared secret key K;
3) The scrambling process is realized by adding the scrambling sequence to the chip-rate spread signal.

As described in [4], [11], the shared secret data between the mobile station and base station can be updated from time to time. To prevent malicious key reload, the key update request can only be initiated from the base station.

IV. Security of the Proposed Scrambling Process

In this section, we use Data Encryption Standard (DES) [7] as a benchmark to evaluate the security of the proposed secure scrambling, which is essentially ensured by AES. We compare the number of possible keys of AES and that of IS-95 scrambling sequence. The number of keys determine 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 exhaustive key search. 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 and could recover a DES key after a few hours. In other words, by trying all possible key values, the hardware could determine which key was used to encrypt a message [2]. Compared with DES, IS-95 has only 42-bit shared secret. The approximate number of keys is about 4.40×10^{12}, which is less than 10^{-4} of the number of DES 56-bit keys. This makes it possible to break the IS-95 long code mask almost in real time through exhaustive key search.

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

- 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, if we choose $L = 128$, then 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.

Security measurement through the number of all possible keys is based on the assumption that the attacker has no easy access to the secret encryption key, therefore, the attacker has to perform an exhaustive key search in order to break the system. As is well known, the security of AES is based on the infeasible complexity in recovering the encryption key. Currently, no weakness has been detected for AES, thus, exhaustive key search is still being recognized as the most effective method in recovering the encryption key and breaking the cryptosystem. In our case, in order for the attacker to obtain the scrambling sequence, the attacker needs to know the input sequence and encryption key. It is reasonable to require that the 42 bits initial secret of the LFSR in Figure III be kept secret together with the 128 bits encryption key. And the attacker will only have access to the scrambled message sequence, for which the secure scrambling sequence is generated from encryption of a 128-bit segment of the LFSR sequence using 128-bit shared secret key between the mobile station and the base station.

As pointed out in Section 2, for the IS-95 system, the entire scrambling sequence can be regenerated as long as 42 successive bits of the scrambling sequence are recovered. In the proposed procedure, even if one block of the scrambling sequence is intercepted, the attacker still needs to recover the secret key K and the input segments $[s_{t+id} \cdots s_{t+127+id}]$ in order to regenerate the entire scrambling sequence, that is, the attacker still needs to break AES.

The key update technique currently used can reduce the risk for the opponent to maliciously reload a new key since the process is controlled by the base station. However, it is still essential to protect the encryption key and to protect the mobile station from being hacked by the malicious attackers.

V. Performance of CDMA Systems with Secure Scrambling

Pseudo-random scrambling in CDMA systems provides physical layer built-in user privacy for information transmission. However, from communication point of view, scrambling was originally designed to reduce interference of mobiles that use the same channelization code in different cells, and to ensure performance stability among user population by providing the desired wide-band spectral characteristics, since the Walsh functions may not spread each symbol's power spectrum uniformly in the available frequency band [8], [10]. When applying secure scrambling, two natural questions are:

1) What effect does it have on system performance?
2) Will it introduce significant computational complexity?

In this section, it will be demonstrated that while providing strong physical layer built-in security, secure scrambling has comparable computational complexity and system performance with that of the conventional scrambling process.

First, we compare the computational complexity of the proposed secure scrambling and conventional scrambling. For this purpose, we only need to compare the complexity of the two scrambling sequence generation methods. Note that they both use the same 42-bit LFSR as specified in (1). In IS-95, each bit of the long scrambling code is generated through

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

For the proposed secure scrambling, every 128-bit block of the scrambling sequence is generated through one AES encryption process. Using a Dell computer with 1024M RAM and 2.8GHz CPU speed, the result is provided in Table 1. As can be seen, the computational complexity of secure scrambling is comparable with that of the scrambling process used in IS-95.

Method	Time required for every 128 bits
IS-95	0.0226 second
Secure scrambling	0.0536 second

TABLE I
Complexity comparison of the two generation methods of long scrambling sequences

Next, under the same spectral efficiency, we compare the input-output BER (bit-error-rate) performance of CDMA systems with conventional scrambling and secure scrambling, respectively. In practical systems, after spreading and scrambling, passband PAM (pulse amplitude modulation) is performed. Mapping information bearing bits to symbols, passband PAM is equivalent to a complex-valued baseband PAM system [9] . When BPSK or QPSK is chosen, the modulo two addition between the message bits and the spreading sequence or the scrambling sequence is now equivalent to multiplying the message symbols using binary (± 1) sequences. In this paper, our discussion is based on the equivalent discrete-time baseband PAM model of CDMA systems, for which the spreading sequences and scrambling sequences are both binary antipodal sequences.

Consider a DS-CDMA system with M users and K receive antennas. Assuming the processing gain is N, that is, there are N chips per symbol. Let $u_j(k)$ ($j = 1, \cdots, M$) denote User j's kth symbol. Without loss of generality, let

$$\mathbf{c}_j = [c_j(0), c_j(1), \cdots, c_j(N-1)] \tag{8}$$

denote User j's channelization code or spreading code. The spread chip rate signal can be expressed as

$$r_j(n) = \sum_{k=-\infty}^{\infty} u_j(k)c_j(n-kN). \tag{9}$$

The successive scrambling process is achieved by

$$s(n) = r_j(n)d_j(n), \tag{10}$$

where $d_j(n)$ is the chip-rate scrambling sequence of user j.

Let $\{g_j^{(i)}(l)\}_{l=0}^{L-1}$ denote the (chip-rate) channel impulse response from jth user to ith antenna, the received chip-rate signal at the ith antenna ($i = 1, 2, \cdots, K$) can be expressed as

$$y_i(n) = \sum_{j=1}^{M}\sum_{l=0}^{L-1} g_j^{(i)}(l)s_j(n-l) + w_i(n). \tag{11}$$

where $w_i(n)$ is the additive noise.

Based on (11), desired user's signal can be extracted through a two-stage procedure. First, training based channel estimation is performed through correlation. Secondly, Rake receiver is applied to combine multipath components. It should be pointed out that currently, it is a common practice in industry to choose the chip rate training sequence be all 1's. The training sequence is put as a prefix to the the chip rate message sequence, and then scrambled using the long scrambling sequence. Channel estimation is therefore carried out based on the correlation property of the front part of the scrambling sequence.

This practice has two drawbacks. First, from security point of view, the front part of the scrambling sequence is exposed to attackers, which makes it possible to recover the whole scrambling sequence right away if secure scrambling is not used. This, at the meantime, illustrates the importance of secure scrambling, which can prevent the whole scrambling sequence being recovered based on the knowledge of part of it. Secondly, from the performance point of view, the correlation property of part of the scrambling sequence may not be ideal, and it can decrease the system performance due to non-accurate channel estimation.

To overcome these shortcomings, we proposed to scramble the training sequence with an independent short scrambling sequence. The training sequence and its scrambling sequence are designed subject to the following constraints:

1) The short scrambling sequence is independent of the long scrambling sequence.
2) The short scrambling sequence has the same length as that of the training sequence.
3) The scrambled training sequence is a Gold sequence.

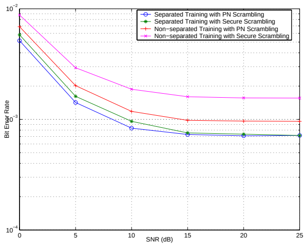

Fig. 4. BER versus SNR, processing gain N = 16, number of user = 4

Or equivalently, we can choose the training sequence be a Gold sequence and then no scrambling is necessary for it. At the meantime, the information sequence is scrambled with the long scrambling sequence. In other words, training sequence is separated from the information sequence in the scrambling procedure. As a result, the long scrambling sequence will not be exposed to malicious attackers and the channel estimation can be performed based on the low cross-correlation of Gold sequences. We term the proposed approach as "*separated training*", and denote the conventional practice by "non-separated training".

In the simulation, we choose the processing gain be $N = 16$, and consider the single receiver case. It is assumed that QPSK signals are transmitted over four-ray multipath channels for each user, with the first path be the dominant path. The multipath delays are uniformly distributed over the interval $[0, N-1]$. That is, the maximum multipath delay L is allowed to be up to one symbol period, a reasonable assumption for wideband CDMA systems. The short scrambling sequence is chosen to be Gold sequences of length 63, and training sequence is chosen to be a sequence of all 1's of the same length. Without loss of generality, User 1 is chosen to be the desired user. Figure 4 shows the bit-error-rate (BER) versus different signal-to-noise ratio (SNR) levels, assuming 4 equal power users in the system. SNR is defined as the chip SNR with respect to User 1. Multipath channels and information sequence consists of 1024 QPSK symbols are generated randomly in each Monto carlo run. And the result is averaged over 100 runs.

As can be seen, system with secure scrambling has comparable performance with that of IS-95, and "*separated training*" delivers much better results compared to that of "non-separated training".

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 mask and a 42-bit

LFSR as in IS-95, the scrambling sequence is generated through AES operations. As a result, the physical layer built-in security of the CDMA system is significantly increased with very limited complexity load. Moreover, it is shown that by scrambling the training sequence and the message sequence separately with two independent scrambling sequences, both information privacy and system performance can be improved. The proposed scheme can readily be applied to 3G systems and IEEE 802.11b WLAN systems.

References

[1] S. Bhashyam and B. Aazhang. Multiuser channel estimation and tracking for long-code CDMA systems. *IEEE Trans. on Communications*, 50(7):1081–1090, July 2002.

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

[3] C.J. Escudero, U. Mitra, and D.T.M. Slock. A Toeplitz displacement method for blind multipath estimation for long code DS/CDMA signals. *IEEE Trans. on Signal Processing*, 49(3):654 –665, March 2001.

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

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

[6] James L. Massey. Shift-Register Synthesis and BCH Decoding. *IEEE Trans. on Information Theory*, 15:122–127, January 1969.

[7] National Bureau of Standards. DES modes of operation. Technical Report FIPS Publication 81, National Bureau of Standards, 1980.

[8] S. Parkvall. Variability of User Performance in Cellular DS-CDMA–Long versus Short Spreading Sequences. *IEEE Trans. on Communications*, 48(7):1178–1187, July 2000.

[9] J.G. Proakis. *Digital Communications*. McGraw-Hill, 4th edition, 2000.

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

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

[12] Lang Tong, van der Veen A., P. Dewilde, and Youngchul Sung. Blind Decorrelating RAKE Receivers for Long-Code WCDMA. *IEEE Trans. on Signal Processing*, 51(6):1642 –1655, June 2003.

[13] A.J. Weiss and B. Friedlander. Channel Estimation for DS-CDMS Downlink with Aperiodic Spreading Codes. *IEEE Trans. on Communications*, 47(10):1561–1569, October 1999.

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

Appendix: A Brief Introduction to AES Algorithm

AES is a secret key block cipher. Namely, it breaks the plaintext into blocks and encrypts each block separately. Three different block sizes are supported in AES: 128 bits, 192 bits and 256 bits with three allowable encryption key sizes: 128 bits, 192 bits and 256 bits. Here, for simplicity, the block size and key size will both be limited to 128 bits.

Let M denote the 128 bits plaintext sequence to be encrypted. At the beginning of the cipher, M is divided into 16 continuous bytes

$$M = [m_0, m_1, \cdots, m_{15}].$$

These 16 bytes are then arranged into a 4×4 matrix and is copied to a 4×4 array $a_{i,j}, i, j = 0, 1, 2, 3$, called the *State Array*, as follows:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \triangleq \begin{bmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{bmatrix}.$$

In AES cipher, the following four basic steps (also called layers), the ByteSub Transformation, the ShiftRow transformation, the MixColumn transformation and the AddRoundKey transformation are defined to form a round. To ensure strong security while minimizing the implementation complexity, ciphers are generated by repeating the same process module (called a round) multiple times. For AES with block size and key size equal to 128 bits, the number of rounds N_r is chosen to be 10 in the standard.

1) **ByteSub Transformation** This layer operates on each byte of the State Array matrix independently using a substitution table, called S-box, please refer to [5]. To do this, each entry in the State Array matrix is divided into two 4-bit groups and written as two hexadecimal numbers X, Y and $a_{i,j}$ is then substituted by the entry of the S-box at row X and column Y. The output of the ByteSub is again a 4×4 matrix of bytes, denoted as

$$B = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}.$$

2) **ShiftRow Transformation** In the ShiftRow transformation, the bytes in the last three rows of the State Array matrix B are cyclically shifted left by 1, 2, and 3 positions respectively to obtain

$$C = \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \triangleq \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,1} & b_{1,2} & b_{1,3} & b_{1,0} \\ b_{2,2} & b_{2,3} & b_{2,0} & b_{2,1} \\ b_{3,3} & b_{3,0} & b_{3,1} & b_{3,2} \end{bmatrix}.$$

3) **MixColumn Transformation** At this step, regarding each bytes $c_{i,j}$ in C as an element of $GF(2^8)$ and multiply the 4×4 matrix C by a matrix with entries in $GF(2^8)$, represented in hexadecimal, to produce

$$\begin{aligned} D &= \begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} \\ &\triangleq \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}. \end{aligned} \tag{12}$$

4) **AddRoundKey Transformation** In this step, a round key matrix, derived from the encryption key (please refer to [5] for *AES Key Schedule* description), is added to the State Array D by a simple bitwise XOR operation.

$$\begin{aligned} E &= \begin{bmatrix} e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\ e_{1,0} & e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,0} & e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,0} & e_{3,1} & e_{3,2} & e_{3,3} \end{bmatrix} \\ &\triangleq \begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} \oplus \\ &\quad \begin{bmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix}. \end{aligned} \tag{13}$$

This is the final output of the round.