Paper: ASAT-15-161-CE

15th International Conference on
***AEROSPACE SCIENCES & AVIATION TECHNOLOGY***,
***ASAT - 15*** – May 28 - 30, 2013, Email: asat@mtc.edu.eg ,
Military Technical College, Kobry Elkobbah, Cairo, Egypt,
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

# Spread Spectrum Encryption Architecture SSEA: A New Encryption Architecture for Post Quantum Computing - Design and Analysis.

{M. H. Megahed[*], Dimitrios Makrakis[†], H. Mouftah[‡], Carlisle Adams[§]}[**]

**Abstract:** The fast development towards building Quantum Computer (QC) increases the consequences of QC attacks and implies high vulnerabilities to symmetric key cipher systems and public key cipher systems. Increasing key length for symmetric key cipher systems to resist QC attacks implies increasing design size of the algorithm which means slow down the algorithm. Inspired from the unpredictability principle, PRNG is added to the architecture of the symmetric key cipher system to add the unpredictability property to choose which algorithm is used and which subkey is used. Spread Spectrum Encryption Architecture (SSEA) is a family of three architectures with high security level and high speed resistant to QC attacks. First, SSEA has two or more encryption algorithms and multiple subkeys at each round of the encryption algorithm. SSEA architecture is used to hide which algorithm is used, to hide which subkey is used and to hide the output of the encrypted ciphertext. Second, SSEA security level is increased as the number of subkeys for each round increased or the number of rounds in the algorithm increased or the number of algorithms increased. This model increases the security level where the output from the PRNG is not on the communication channel and the attacker cannot perform analysis to this output. Third, cryptanalysis cannot take place over SSEA; the only way for the attacker to break SSEA is to establish brute force attack over all of the system possible combinations. Finally, SSEA3 is chosen to be implemented as it has the highest speed, the lowest design size and the highest security level over SSEA1, SSEA2 and AES-256 full rounds.

**Keywords:** Unpredictability; PRNG; key schedule; subkeys; spread spectrum encryption architecture; SSEA; quantum computer.

## 1. Introduction

Now, new classification for cryptography is emerged after the formal modern cryptography: Pre-Quantum Computing and Post-Quantum Computing. This is because quantum computer enables certain problems to be solved efficiently in short time. We will prove that QC cannot improve on classical methods to solve unpredictability problem that we based our new designed architecture on it. Even the QC needs to try all the possible combinations to solve unpredictable problem. Quantum computation will have significant impact on symmetric key cipher systems and public key cipher systems.

[*] mmega080@uottawa.ca
[†] Prof., dimitris@site.uottawa.ca
[‡] Prof., mouftah@uottawa.ca
[§] Prof., cadams@site.uottawa.ca
[**] School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada.

In 1994, Peter Shor has presented quantum algorithms which solve the factoring and the discrete logarithm problems in quantum polynomial time [**1**]. These problems are very difficult in the classical computer model and they provide a basis for the security of the most currently-used public key cryptosystems.

In 1996, Lov Grover has developed a quantum algorithm for searching an unsorted database with $N$ entries in $O(\sqrt{N})$ time and using O(*log N*) storage space [**2**-**3**]. As a result, the brute force attack on symmetric cipher systems can be obtained in only $O(\sqrt{N})$ steps instead of $O(N)$. If a suitably sized quantum computer capable of running Grover's algorithm reliably becomes available, it would reduce a 128-bit key down to 64-bit security, roughly a DES equivalent. This is one of the reasons why AES supports a 256-bit key length. Also, Bennett, Bernstein, Brassard, and Vazirani proved in 1996 that a brute-force key search on a quantum computer cannot be faster than roughly $2^{n/2}$ invocations of the underlying cryptographic algorithm, compared with roughly $2^n$ in the classical case [**4**]. Thus in the presence of large quantum computers an $n$-bit key can provide at least $n/2$ bits of security. Quantum brute force is easily defeated by doubling the key length, which has extra computational cost in ordinary use. This implies that at least a 256-bit symmetric key is required to achieve 128-bit security rating against a quantum computer.

On 2004, the eSTREAM, ECRYPT (European Network of Excellence for Cryptology) Stream Cipher Project, starts. This four years effort running from 2004 to 2008 has identified two portfolios of promising new stream ciphers, one for software orientation and the other for hardware orientation. The eSTREAM raised a question, if large QC can be built, how will this influence the symmetric key cryptographic landscape? [**5**].

In [**6**], Akihiro Yamamura and Hirokazu Ishizuka on 2000 were the first who discussed how to attack the block cipher algorithms with multiple of QCs using the Grover's algorithm.

In [**7**], Gilles Piret and François-Xavier discussed on 2009 the distance between the practical security approach and the actual theoretical security provided by a given cipher. Their experiments illustrated that the provable security against linear cryptanalysis is not achieved by present design strategies and the relevance of the practical security approach. Finally, they discussed the impossibility to provide provable security of block ciphers against linear cryptanalysis.

Now, the existing proposed solution is to increase the key length with increasing the design size which will result in slow down the algorithm. Our new designed SSEA has a significant security level of an exponential gain above all other existing encryption architectures as the number of subkeys for each round increased or the number of rounds increased or the number of algorithms increased. Our proposed solution can mitigate all cryptanalysis attacks to encryption.

### 1.1 Motivations

*First*, we reviewed some quantum algorithms, some quantum applications and the advancements to build QC to know what QC can do and what it cannot do. We found that QC is same as classical computer when solving the problem of unpredictability to try all possibilities to find the solution. This background helped us to develop our new computationally unbreakable SSEA. *Second*, we found that increasing the key length for symmetric key cipher systems implies additional design size which slows down the encryption process as AES-256 is 14 rounds compared to AES-128 which is 10 rounds. We need high

speed encryption architecture with high security level resistant to QC attacks. We found that unpredictability will result in reducing the number of rounds in block cipher using SSEA. *Third*, we need a cryptographic system that is efficient on multiple platforms in both hardware and software implementations. SSEA is efficient in both hardware and software implementation. *Fourth*, we need a cryptographic system with high security level. SSEA security level is increased exponentially as number of algorithms increased or number of rounds increased or number of subkeys for each round increased.

### 1.2 Outline of the Paper

In *section 2*, the preliminary information from multiple discipline areas are given. These preliminaries are the hypothesis of the design, the design goals, dynamic security and unpredictability principal. *Section 3* discussed the threat model. *Section 4* presented the existing solutions for symmetric key ciphers to resist QC attacks. *Section 5* outlined our new designed SSEA. *Section 6* explained the proof of security for SSEA3. *Section 7* discussed the attacks to SSEA3. *Section 8* compared between our new designed SSEA3 and the standard AES-256. *Section 9* concluded the results. Finally, *section 10* briefly outlined our future research works.

### 1.3 Contributions

1- We developed a strong barrier for QC which is unpredictability to find the right algorithms sequence, the right subkeys sequence and the right stream of bits to encrypt the ciphertext before starting cryptanalysis.
2- We developed the SSEA family.
3- We developed the first encryption architecture characterized by increasing the security level exponentially with increasing the number of subkeys at each round.
4- We developed the first encryption architecture characterized by increasing the security level exponentially with increasing the number of encryption algorithms.
5- We developed the first encryption architecture characterized by increasing the security level exponentially with increasing the number of rounds in the algorithm.
6- SSEA has reduced rounds AES-256 but higher security level.

## 2. Preliminaries

Most encryption architectures, i.e. using multiple encryption algorithms, are depending on fixed architecture therefore, cryptanalysis can be performed over these architectures. In order to mitigate QC attacks and cryptanalysis attacks, we added the property of unpredictability to the encryption architecture as we designed a key-dependent encryption architecture that uses at least two algorithms, at least 16 subkeys at each round and at least 3 rounds of AES-256. We named the algorithm spread spectrum encryption architecture because the encryption is done by more than one algorithm and more than one subkey at each round then we mask the output ciphertext with encrypted stream of bits.

### 2.1 Hypothesis of the Design

It is very hard to mitigate QC attacks and cryptanalysis with high speed and high security level encryption architecture but adding unpredictability to encryption architecture through exploring the capabilities of dynamic encryption approaches [**8**-**9**] for cryptography, will help to build strong architecture resistant to QC attacks and cryptanalysis . We started our design by assuming that we need high security level and high speed encryption algorithm.

*In SSEA1*, we will use two algorithms of AES-256 with two different S-Boxes each algorithm has 7 rounds. We will use RC4 stream cipher algorithm to choose which algorithm is used to encrypt the data. *In SSEA2*, we will use one algorithm of AES-256. We will use 16 subkeys at each round of the encryption algorithm and the algorithm has 7 rounds. We will use RC4 stream cipher algorithm to choose which subkey is used at each round to encrypt the data. *In*

*SSEA3*, we will use two encryption algorithms of AES-256 with two different S-Boxes each algorithm has 3 rounds. We will use RC4 stream cipher algorithm to choose which algorithm is used to encrypt the data and the other algorithm will have 128 bits input from the RC4 output and the two outputs from the two algorithms are XORed. SSEA architecture overcomes the weakness of fixed key and fixed algorithm architecture and protects wireless network against cryptanalysis. The attacker cannot obtain the output sequence from the PRNG to analyze it; therefore, the attacker must start the cryptanalysis for all possible combinations of the subkeys groups and encryption algorithms sequences.

## 2.2 Goals of the Design

In this paper we have five goals to achieve as follows:

1- Computationally Unbreakable: We designed this encryption architecture to make it very hard to break a long piece or even a short piece of ciphertext without the knowledge of the output of PRNG.
2- Implementable in both Software and Hardware: The SSEA can work perfectly without any constraints from software or hardware perspectives.
3- Controlling the Security Level: The SSEA increases the security level exponentially each time a subkey is added to the system or a round is added to the system or an algorithm is added to the system.
4- Prevent attacker from applying chosen plaintext ciphertext attack: The SSEA can prevent the attacker from applying chosen plaintext ciphertext attack because the attacker does not know the plaintext will go to algorithm one or algorithm two also, the attacker has no clue the plaintext is encrypted with which subkeys.
5- High Speed Algorithm: SSEA3 has 3 rounds AES-256 compared to 14 rounds AES-256.

## 2.3 Dynamic Encryption

Dynamic encryption can be achieved by four main categories which are the followings:

1- Key dependent components such as S-Boxes and permutations. At the start of the secure session, we fill the S-Boxes in the encryption algorithm such as Twofish encryption algorithm
2- Configuration of encryption components to choose one component from multiple components or to choose one encryption algorithm from multiple algorithms. At the start of the secure session we choose the used S-Box component from multiple S-Boxes or we choose the encryption algorithm from multiple algorithms such as IPSec and SSL.
3- Spread Spectrum Encryption Architecture (SSEA).

Finally, we will use our designed SSEA to apply the dynamic security.

## 2.4 Unpredictability Principle

If we use a cipher that includes a general computational process sequence, and keep all the sequence of computations of that process secret, the cryptanalyst will face a problem which he will be unable to solve except by trying all possible combinations. We found that static encryption systems are deterministic and they are susceptible to cryptanalysis but dynamic encryption systems need dynamic cryptanalysis process which is an obstacle to cryptanalysis. The output controlling sequence from the RC4 to choose the subkeys for each round or to choose one algorithm from multiple algorithms is unknown to the attacker and this provides the SSEA with the unpredictability principal where the subkeys keep changed for every plaintext block and the used algorithm keeps changed for every plaintext block. Unpredictability stops cryptanalysis where the only way to break the system is by using brute force attack and by trying all possible combinations of the output RC4.

# 3. Threat Model

There are many factors which work together to compromise the security of the symmetric key cipher systems; these are the cryptanalysis techniques, supercomputer, Quantum Computer, side channel attacks, grid computing, parallel processing, and the special purpose hardware for cryptanalysis such as COPACOBANA embedded system [**10**]. Therefore, there are increasing demands to design high speed and high security level new encryption architecture resistant to all these attacks. We suppose that our system adversary is the QC that implements Grover's algorithm to find the used key for every ciphertext block. Also, we suppose that a supercomputer is trying to cryptanalyze our system.

# 4. Existing Works

Existing security systems in communication systems or in computer networks relay on a set of encryption algorithms which are secure until cryptanalysts break them. These existing security schemes are vulnerable to cryptanalysis techniques; therefore, there are high demands to provide a barrier between the encryptor unit and the growing attacks from cryptanalysis.

**Today's Existing Encryption Architectures:**

1- Survivable security architecture using multiple encryption algorithms as IPSec and SSL protocols.
2- Cascaded encryption architecture using two or three encryption algorithms.
3- Compression then encryption architecture.
4- Proactive security architecture through frequently changing the key.
5- Using feedback modes of operations for block cipher encryption algorithms.
6- Key-dependent components architecture as S-Boxes.
7- Stream cipher controlling block cipher key-schedule architecture [**11**].

All the mentioned encryption architectures have fixed architecture except the key dependent component architecture but it can be cryptanalyzed by large number of known plaintext ciphertext pairs. Cryptographic experts recommend increasing symmetric key cipher systems key length to be 256 bits key length to resist QC upcoming attacks but cryptanalysis is still applicable for AES-256.

# 5. Overview of SSEA

## 5.1 SSEA Family

If we need a barrier between the encryption algorithm and the cryptanalysis, the SSEA is the perfect barrier. If we want stronger security guarantee, we need to add unpredictability to cryptosystem and this is done for SSEA. SSEA is a family of three architectures for symmetric key cipher systems. SSEA1 architecture is concerned with choosing one algorithm from multiple algorithms. SSEA2 architecture is concerned with choosing one subkey from multiple subkeys at each round of the block cipher algorithm. SSEA3 architecture is concerned with choosing one algorithm from multiple algorithms, choosing one subkey from multiple subkeys at each round of the block cipher algorithm and masking the output ciphertext with encrypted stream of bits. The three architectures are dynamic and the third one is the strongest one.

## 5.2 SSEA1 Architecture

### 5.2.1 System components

1- Two AES-256 Encryption algorithms with 7 rounds.

   We use two AES-256 encryption algorithms with two different S-Boxes to solve the synchronization problem between the two used algorithms. Different S-Boxes ensure different output with same key.

2- Key schedule.

   There are two keys of 256 bits key length. We choose the key schedule of AES-256 to generate all subkeys of the two AES-256 encryption algorithms and the 256 bits seed for the RC4 stream cipher algorithm.

3- RC4 stream cipher algorithm as PRNG.

   We use RC4 stream cipher algorithm as PRNG for the architecture.

### 5.2.2 Encryption

Figure 1 shows the SSEA1 architecture which is composed of two AES-256 encryption algorithms with two different S-Boxes such as S1 and S2. Each algorithm has only 7 rounds not 14 rounds this is because the 7 rounds AES-256 needs $2^{32}$ chosen plaintext ciphertext pairs to break the 7 rounds [**12**]. Each pair has two possibilities to enter algorithm one or algorithm 2 and 10 pairs has $2^{10}$ possible combinations. Therefore, $2^{32}$ pairs has ($2\hat{}2^{32}$) possible combinations which is infeasible to try by the attacker. The PRNG chooses which algorithm is used to encrypt the plaintext. The sequence of PRNG output is not on the communication channel and this fact is the most glamour property of SSEA1 to prevent the attacker from knowing the sequence of using the encryption algorithms. The plaintext enters all the encryption algorithms to stop side channel attack but we choose the output ciphertext according to the PRNG output which is only known to the receiver. For simplicity, SSEA1 has two encryption algorithms and it can have more than two encryption algorithms.

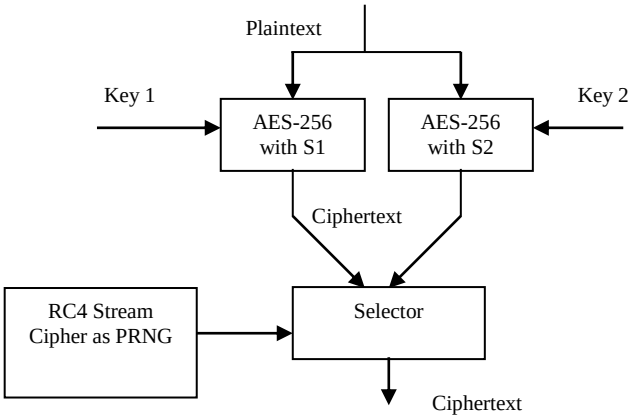


**Fig.1 SSEA1 Architecture with two encryption algorithms**

### 5.2.3 Decryption

The PRNG chooses which algorithm is used to decrypt the ciphertext. The sequence of PRNG output is not on the communication channel to prevent the attacker from knowing the sequence of using the encryption algorithms. The ciphertext enters all the encryption algorithms but we choose the output plaintext according to the PRNG output which is only known to the receiver. Allowing all encryption algorithms to decrypt will prevent side channel attack.

### 5.2.4 Mathematical model

*For Encryption:*

$C_j = \{E_i (P_j)_{Ki}$ under $S_j\}$

The ciphertext is a function of two inputs which are the plaintext and the PRNG output.

$C_j$ is the ciphertext where j = 1 to n and n is the number of plaintexts, $E_i$ is the encryption algorithm and we have two encryption algorithms where i = 1 or 2, $P_j$ is the plaintext, $K_i$ is the key of the encryption algorithm and we have two keys for the two encryption algorithms, $S_j$ is one bit from RC4 stream cipher algorithm as PRNG. $S_j$ selects one algorithm output to be the ciphertext.

*For Decryption:*
$P_j = \{D_i (C_j)_{Ki}$ under $S_j\}$
$D_i$ is the encryption algorithm and we have two encryption algorithms where i = 1 or 2, $S_j$ selects one algorithm output to be the plaintext.

### 5.2.5 System analysis

The only way for the attacker to break the system is to try all possible combinations which is (2^P) where P is the number of plaintext blocks. (2^P) is infeasible to try by the attacker if P exceeds 256. This is because the PRNG is not on the communication channel. The algorithm architecture stops the cryptanalysis because the attacker does not know which algorithm was used to encrypt the plaintext. The attacker needs to try all possible combinations to know the PRNG output sequence.

Finally, SSEA1 with two AES-256 encryption algorithms which have two different S-Boxes is a strong barrier against QC attacks and it has slightly larger design size as AES-256 because we use only seven rounds of AES-256 and RC4 stream cipher algorithm. SSEA1 has higher speed than AES-256 and higher security level and there is no cryptanalysis technique which can break SSEA1. Since the encryption architecture is dynamic, the attacker cannot perform linear and differential cryptanalysis.

### 5.2.6 SSEA1 advantages

1- The attacker cannot apply known plaintext ciphertext attack or chosen plaintext ciphertext attack to the encryption architecture because the attacker does not know the ciphertext came from algorithm one or algorithm two.

2- The attacker cannot apply cryptanalysis techniques such as linear and differential cryptanalysis to the encryption architecture because the attacker does not know the ciphertext came from algorithm one or algorithm two.

3- The only way to attack the system is brute force attack which needs to guess 512 bits key length and to try all possible combinations of using two algorithms and this is impossible.

4- We can use reduced rounds AES-256. Therefore, we use seven rounds AES-256 which needs $2^{32}$ chosen plaintext ciphertext pairs to break the algorithm with total of ($2^{\wedge}2^{32}$) possible combinations which is infeasible to try by the attacker.

5- SSEA1 has higher speed than AES-256.

6- SSEA1 has higher security level than AES-256.

7- SSEA1 has the higher key length than AES-256 which is 512 bits.

### 5.2.7 SSEA1 disadvantages

1- The architecture has larger design size by using RC4 stream cipher algorithm as PRNG and two AES-256 reduced rounds algorithm each of seven rounds.

2- The architecture needs extra synchronization cost to synchronize the two RC4 algorithms at transmitter and receiver.

### 5.2.8 SSEA1 cryptanalysis

We cannot use AES-128. Attacker can get the 128 bits key from one known ciphertext plaintext pair using QC.

There is no need to use full rounds AES-256. We need $2^{32}$ known plaintext ciphertext pairs [**12**] to break seven rounds AES-256. These pairs require ($2^{\wedge}$ $2^{32}$) possible combinations which is infeasible to try by the attacker.

## 5.3 SSEA2 Architecture

### 5.3.1 System components

1- One AES-256 Encryption Algorithm with 7 rounds.

We use AES-256 as the encryption algorithm. We use only seven rounds AES-256. This is because we need $2^{32}$ chosen plaintext ciphertext pairs to break the seven rounds algorithm with fixed subkeys [**12**]. The subkeys of SSEA2 are not fixed therefore; we can implement only 7 rounds of AES-256. Each round has 16 subkeys and if there are 7 rounds then we have ($16^7 = 2^{28}$) possible subkeys groups for each plaintext.

2- Key schedule.

There is one key of 256 bits key length. The key schedule generates 16 subkeys from the 256 bits key. We choose one subkey from 16 subkeys at each round from the 7 rounds.

The attacker needs to guess ($16^7 = 2^{28}$) subkeys groups' possible combinations to know the sequence of using subkeys. Each plaintext has $2^{28}$ possible combinations of choosing subkeys groups. The only way for the attacker to break the system is to try all possible combinations which is ($2^{28}$^P) where P is the number of plaintext blocks. ($2^{28}$^P) is infeasible to try by the attacker if P exceeds 9.

The key schedule of AES-256 generates 256 bits seed for the RC4 stream cipher algorithm.

3- RC4 stream cipher algorithm as PRNG.

We use RC4 stream cipher algorithm as the PRNG at each round to choose one subkey from 16 subkeys.

### 5.3.2 Encryption

Figure 2 shows the SSEA2 architecture. The PRNG chooses which subkey is used to encrypt the plaintext. The sequence of PRNG output is not on the communication channel and this fact is the most glamour property of SSEA2 to prevent the attacker from knowing the sequence of using the subkeys. For SSEA2, to encrypt a short message such as 1 plaintext block, the attacker needs to try $2^{28}$ possible combinations to guess the right subkeys. Also, to encrypt a short message such as 100 plaintext blocks, the attacker needs to try ($2^{28}$ ^ 100) possible combinations to guess the right subkeys. Also, to encrypt a long message such as 100,000 plaintext blocks, the attacker needs to try ($2^{28}$ ^ 100,000) possible combinations to guess the right subkeys.

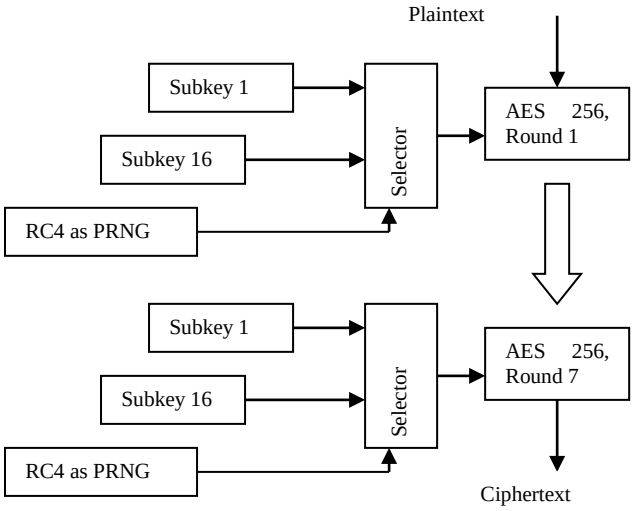


**Fig. 2 SSEA2 encryption architecture**

### 5.3.3 Decryption

The PRNGs choose which subkey is used to decrypt the ciphertext. The sequence of PRNG output is not on the communication channel to prevent the attacker from knowing the sequence of using the subkeys.

### 5.3.4 Mathematical model

*For Encryption:*

$C_j = \{E (P_j)_{Ki}$ under $S_j\}$

The ciphertext is a function of three inputs which are the plaintext, subkeys groups and the PRNG output.

$C_j$ is the ciphertext where j = 1 to n and n is the number of plaintexts, E is the encryption algorithm and we have only one encryption algorithm, $P_j$ is the plaintext, $K_i$ is the subkeys generated for the encryption algorithm and we have ($16^7 = 2^{28}$) subkeys groups for 16 subkeys at each round of seven rounds, $S_j$ is four bits from RC4 stream cipher algorithm as PRNG. $S_j$ selects one subkey from 16 subkeys at each round of 7 rounds.

*For Decryption:*

$P_j = \{D (C_j)_{Ki}$ under $S_j\}$

D is the encryption algorithm and we have only one encryption algorithm, $S_j$ selects one subkey from 16 subkeys at each round of 7 rounds.

### 5.3.5 System analysis

The only way for the attacker to break the system is to try all possible combinations of the PRNG for each ciphertext pair. This is because the PRNG is not on the communication channel. The algorithm architecture stops the cryptanalysis because the attacker does not know which subkeys were used to encrypt the plaintext. The attacker needs to try all possible combinations to know the output sequence of the PRNG.

### 5.3.6 SSEA2 advantages

1- Subkeys are not fixed as they are dynamic.

2- The attacker cannot apply known plaintext ciphertext attack or chosen plaintext ciphertext attack to the encryption architecture because the attacker does not know which subkeys were used to encrypt the plaintext.

3- The attacker cannot apply cryptanalysis techniques such as linear and differential cryptanalysis to the encryption architecture because the attacker does not know the ciphertext is encrypted with which subkeys groups.

4- The only way to attack the system is brute force attack which needs to guess 256 bits key length which is impossible then try all possible combinations of subkeys groups which is ($2^{28}$ ^ P) where P is number of plaintext blocks.

5- We can use reduced rounds AES-256. Therefore, we use 7 rounds AES-256 which needs $2^{32}$ chosen plaintext ciphertext pairs to break the 7 rounds which needs ($2^{28}$ ^ $2^{32}$) possible combinations which is infeasible to try by the attacker.

6- SSEA2 has higher speed than AES-256.

7- SSEA2 has higher security level than AES-256.

8- The encryption design size is lower than AES-256 where we have AES-256 reduced rounds algorithm with 7 rounds and RC4 stream cipher algorithm.

### 5.3.7 SSEA2 disadvantages

1- The architecture needs extra synchronization cost to synchronize the two RC4 algorithms at transmitter and receiver.

2- The architecture needs the 16 subkeys at each round to choose one subkey which is extra cost for hardware.

### 5.3.8 SSEA2 cryptanalysis

We cannot use AES-128. Attacker can get the 128 bits key from one known ciphertext plaintext pair using QC.

There is no need to use full rounds AES-256. Seven rounds AES-256 We need $2^{32}$ known plaintext ciphertext pairs [**12**] to break seven rounds AES-256. These pairs require ($2^{28}$ ^ $2^{32}$) possible combinations which is infeasible to try by the attacker.

## 5.4 SSEA3 Architecture

### 5.4.1 System components

1- Two AES-256 encryption algorithms with 3 rounds.

We use two AES-256 encryption algorithms with two different S-Boxes to solve the synchronization problem between the two used algorithms. Each algorithm has only 3 rounds of AES-256. Different S-Boxes ensure different algorithms output with same key. The encryption algorithm will keep changing from algorithm 1 to algorithm 2. Each round has 16 subkeys of the 3 rounds. The subkeys are not fixed. The attacker needs to guess ($16^3 = 2^{12}$) subkeys groups' possible combinations to know the sequence of using subkeys. Each plaintext has ($2^{12}$) possible combinations of choosing subkeys groups.

Reasons to choose 3 rounds AES-256 for SSEA3:

- We use double encryption. The ciphertext from the plaintext is encrypted with stream of bits came from the second algorithm while the input to the second algorithm only known to the receiver and it is not known to the attacker.
- The subkeys are dynamic and they are changing for every plaintext with ($2^{12}$) possible combinations.
- The architecture is dynamic where the algorithm that encrypts the plaintext is not fixed as we use two encryption algorithms to encrypt the plaintext.
- The attacker cannot perform known plaintext ciphertext attack since the ciphertext is encrypted with unknown input to the attacker.
- The attacker cannot perform man in the middle attack over the ciphertext because the ciphertext is encrypted with unknown input to the attacker.

2- Key schedule.

There are two keys of 256 bits key length. We choose the key schedule of AES-256 to generate 16 subkeys at each round for the 3 rounds of each algorithm. The key schedule of AES-256 generates the 256 bits seed for the RC4 stream cipher algorithm.

3- RC4 stream cipher algorithm as PRNG.

We use RC4 stream cipher algorithm as the PRNG to choose one subkey of the 16 subkeys at each round. The PRNG chooses where the plaintext goes to algorithm 1 or algorithm 2. The output from RC4 stream cipher algorithm is used to enter the algorithm that is not used by the plaintext. The outputs from the two encryption algorithms are XORed.

### 5.4.2 Encryption

Figure 3 shows the SSEA3 architecture with two AES-256 encryption algorithms and two session keys for each algorithm. The RC4 stream cipher algorithm chooses which subkey is used to encrypt the plaintext and the RC4 stream cipher algorithm chooses which algorithm

will encrypt the plaintext and we marked it as output2. The sequence of PRNG output is not on the communication channel and this fact is the most glamour property of SSEA3 to prevent the attacker from knowing the sequence of using the subkeys or the sequence of using the encryption algorithms. The output from the RC4 algorithm is encrypted with the second algorithm and we marked it as output1. The output from the two algorithms is XORed and we marked it as output3. For 3 rounds AES-256, we need $((2^{12} \text{ X } 2^{12} \text{ X } 2 \text{ X } 2^{128})^{\wedge}P)$ possible combinations to break the 3 rounds of the algorithm where P is the number of plaintext blocks. The ciphertext is encrypted therefore, the attacker cannot apply known plaintext ciphertext attack and for this reason we use only 3 rounds AES-256.

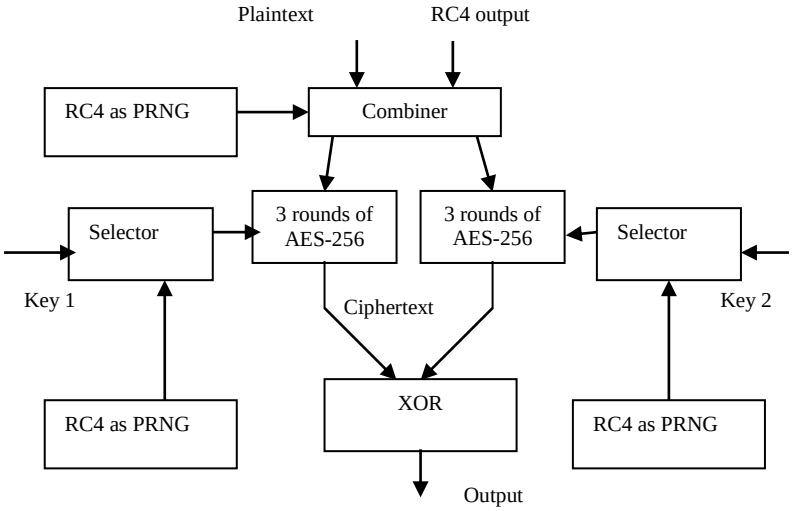


**Fig. 3 SSEA3 Encryption architecture**

### 5.4.3 Decryption

First, we decrypt the output1 from the PRNG and marked it as input1. Second, we perform XOR on the input1 with the output3 from the two encryption algorithms to get the ciphertext of the plaintext which is input2. Third, we decrypt the ciphertext which is input2 with the encryption algorithm that is not used by the RC4 stream cipher algorithm to get the plaintext. The RC4 as PRNG chooses which subkey is used to decrypt the ciphertext. The RC4 as PRNG chooses which algorithm is used to decrypt the ciphertext and which algorithm is used to decrypt the output from the RC4 algorithm. The sequence of PRNG output is not on the communication channel to prevent the attacker from knowing the sequence of using the subkeys or algorithms. The decryption algorithm is double size the encryption algorithm to allow the decryption speed to be the same as the encryption speed.

### 5.4.4 Mathematical model

*For Encryption:*

$C_j = \{E_i (P_j)_{Ki} \text{ XOR } E_i (RC_j)_{Ki} \text{ under } S_j\}$

The ciphertext is a function of three inputs which are the plaintext, subkeys groups and the PRNG output.

$C_j$ is the ciphertext where j = 1 to n and n is the number of plaintexts, $E_i$ is the encryption algorithm and we have two encryption algorithms where i = 1 or 2, $P_j$ is the plaintext, $K_i$ is the subkeys generated for the encryption algorithm and we have $(16^3 = 2^{12})$ subkeys groups for 16 subkeys at each round of 3 rounds, $S_j$ is 141 bits from RC4 stream cipher algorithm to encrypt one plaintext block where we need 4 bits to choose one subkey out of 16 subkeys with total of 12 bits for 3 rounds and one bit to choose one algorithm to encrypt the plaintext and 128 bits to enter the second encryption algorithm. $S_j$ selects one subkey from 16 subkeys at each round of 3 rounds and selects one algorithm to encrypt the plaintext while the other algorithm is

used to encrypt 128 bits output from RC4 stream cipher algorithm. $RC_j$ is the 128 bits output from RC4 stream cipher algorithm.

*For Decryption:*

$RC_j = D_i$ (encrypted $RC_j)_{Ki}$ under $S_j$

$P_j = \{D_i (RC_j$ XOR $C_j)_{Ki}$ under $S_j\}$

$D_i$ is the encryption algorithm and we have two encryption algorithms, $S_j$ selects one subkey from 16 subkeys at each round of 3 rounds and selects one algorithm to decrypt the plaintext while the other algorithm is used to decrypt the 128 bits output from RC4 stream cipher algorithm.

### 5.4.5 System analysis

The only way for the attacker to break the system is to try all possible combinations of the PRNG for each ciphertext pair and for choosing subkeys groups. This is because the PRNG is not on the communication channel. The algorithm architecture stops the cryptanalysis because the attacker does not know which subkey was used to encrypt the plaintext. The attacker does not know the 128 bits output from RC4 that enters the second algorithm. The attacker needs to try all possible combinations to know the output sequence of the PRNG. The attacker does not know the plaintext went to which algorithm. This architecture has much higher speed than AES-256 as it has only 3 rounds. Finally, SSEA3 with AES-256 is strong barrier against QC attacks with higher speed than AES-256 full rounds. The ciphertext is encrypted to prevent linear and differential cryptanalysis.

### 5.4.6 SSEA3 advantages

1- Double encryption.

2- Subkeys are not fixed as they are dynamic.

3- The attacker cannot apply known plaintext ciphertext attack or chosen plaintext ciphertext attack to the encryption architecture because the attacker does not know the ciphertext came from algorithm one or algorithm two and the attacker does not know which subkeys group is used to encrypt the plaintext out of $2^{12}$ subkeys groups.

4- The attacker cannot apply cryptanalysis techniques such as linear and differential cryptanalysis to the encryption architecture because the attacker does not know the ciphertext came from algorithm one or algorithm two and the attacker does not know which subkeys group is used to encrypt the plaintext out of ($2^{12}$ X $2^{12}$) subkeys groups.

5- The only way to attack the system is brute force attack which needs to guess 512 bits key length and to try all possible combinations of using two algorithms and to try which subkeys group is used out of ($2^{12}$ X $2^{12}$) subkeys groups and this is impossible.

6- We can use reduced rounds AES-256. Therefore, we use 3 rounds AES-256. We use 3 rounds because the ciphertext is encrypted and the attacker cannot perform known plaintext ciphertext attacker over SSEA3. SSEA3 needs (($2^{12}$X$2^{12}$X2X $2^{128}$) ^ P) possible combinations where P is the number of plaintext blocks which is infeasible to try by the attacker.

7- The algorithm has higher speed than AES-256.

8- The Algorithm has higher security level than AES-256.

9- The encryption design size is lower than AES-256 where we have two AES-256 encryption algorithms reduced rounds with 3 rounds.

10- The architecture has higher key length than AES-256 which is 512 bits.

### 5.4.7 SSEA3 disadvantages

1- The architecture needs extra synchronization cost to synchronize the two RC4 algorithms at transmitter and receiver.

2- The architecture needs the 16 subkeys at each round to choose one subkey which is extra cost for hardware.

3- The decryption design size is double encryption design size to allow decryption speed to be same as encryption speed but the decryption size is still less than AES-256 full rounds as it has only 12 rounds AES-256.

### 5.4.8 SSEA3 cryptanalysis

We cannot use AES-128. The attacker can get the 128 bits key using QC.
There is no need to use full rounds AES-256. We use 3 rounds AES-256 which require $((2^{12}X2^{12}X2X2^{128})^P)$ possible combinations where P is the number of plaintext blocks which is infeasible to try by the attacker.
We choose to implement SSEA3 because it has the lowest design size, the highest speed and the highest security level.

## 5.5 AES-256 Components

### 5.5.1 AES-256 block cipher encryption algorithm.

The AES is a substitution permutation network (SPN) allowing the encryption/ decryption of data by blocks of 128-bits and supporting key lengths of 128, 192 and 256 bits. In the following, we focus on the 256-bits key version. Its internal state, usually represented as a 4×4 matrix of bytes, is updated by iterating through the round structure (10, 12 or 14 times according to the key size whether 128 or 192 or 256 bits respectively). The round is described as four different byte-oriented transformations [**13**].
*First*, BytesSub introduces the non-linearity by taking, for each byte, the modular inverse in $GF(2^8)$ and then applying an affine transformation. Instead of computing distinctly these two steps, the full transformation is achieved by passing each byte through an S-Box. We use two different S-Boxes for the two AES-256 encryption algorithms of SSEA1 and SSEA3.
*Second*, ShiftRows modifies the state. It simply consists of a circular left shift of the state's rows by 0, 1, 2 and 3 bytes respectively.
*Third*, MixColumns applies a linear transformation to the state's columns. Each of them is regarded as a polynomial and is multiplied by a fixed polynomial
$c(x) = 3x^3 + x^2 + x + 2 \pmod{x^4 + 1}$.
*Finally*, the AddRoundKey transform mixes the key with the state. As each subkey has the same size as the state, the combination is performed by a simple bitwise XOR between subkey bytes and their corresponding state bytes as shown in Figure 7. A first key addition is performed before entering the first round, and the last round omits the MixColumns transformation.

### 5.5.2 Block cipher key schedule.

Prior to the encryption/decryption process, the subkeys have to be generated. The key schedule takes the main key $K_0$ and expand it for the case of a 256-bit key, where SubWord applies the S-Box to the 32-bit input word, RotWord rotates the word one byte to the left and RC(i) is an 8-bit constant associated to each round i.

## 5.6 RC4 Stream Cipher Algorithm

We use RC4 stream cipher algorithm as PRNG to stop the used PRNG from entering dead lock state. Dead lock state is the state of continued all '0' state.

## 6. SSEA Proof of Security

Our designed SSEA3 applied Kerckhoffs' Principle which stated that "A cipher should be secure when the cryptanalyst knows all details of the enciphering process and deciphering process except the value of the secret key". The cryptanalyst knows everything about the encryption algorithms and the PRNG generates the controlling sequence, except the algorithms secret keys and the PRNG controlling sequence.

Shannon distinguished between two types of security:

- Unconditionally secure - means security against an enemy who has unlimited time and computational resources.
- Computationally secure - means security against an enemy who has a specified limited amount of time and computational resources.

SSEA3 cipher is computationally secure which means it cannot be broken with the current computer technology within limited time and computational resources.

**Definition 1:**

Let $S$ be the output controlling sequence of PRNG, let $A_1$ and $A_2$ be the used block cipher algorithms with n rounds which uses at least 16 subkeys at each round, let $L_{11}$ to $L_{16}$ be 16 subkeys of the key, let the controlling sequence at each round from the encryption algorithm chooses one subkey from the 16 subkeys, let the controlling sequence choose which algorithm is used to encrypt the plaintext and let the controlling sequence enters the second algorithm to mask the ciphertext by XORing the ciphertext with the output from the second algorithm. Every plaintext block is encrypted with a different group of $n$ subkeys. Since the output of the PRNG is not on the communication channel and the attacker cannot analyze it therefore, SSEA3 is computationally secure where there are number of subkeys groups equal to 16^$r$ and r is number of rounds.

The PRNG uses a seed to generate the control sequence. The receiver must use the same PRNG with the same seed to generate the same sequence to be able to decrypt the ciphertext. This architecture is a strong barrier for cryptanalysts to break. Therefore, SSEA3 can be used for Post-Quantum Computing to resist QC attacks.

**Definition 2:**

We can define the computational security as follows [**14**]:

Let ($E$;$D$) be an encryption / Decryption scheme that uses $n$-bit keys to encrypt $\ell(n)$-length messages.

(E;D) is computationally secure if for every polynomial-time algorithm $A$: $\{0,1\}^* \rightarrow \{1,0\}$, polynomially bounded $\varepsilon$: $\{0,1\}^* \rightarrow [1,0]$, $n$, and $x_0$, $x_1$ is subset of $\{0,1\}^{\ell(n)}$ ,

$\left| Pr[A(E_{Un}(x_0)) = 1] - Pr[A(E_{Un}(x_1)) = 1] \right| < \varepsilon(n)$.

Traditional cryptosystem is five tuples ($P$, $C$, $K$, $E$, $D$), where $P$ is the plaintext, $C$ is the ciphertext, $K$ is the key space, $E$ is the encryption algorithm, and $D$ is the decryption algorithm.

SSEA is six tuples ($P$, $C$, $K$, $E$, $D$, $S$), where $P$ is the plaintext, $C$ is the ciphertext, $K$ is the key space, $E$ is the encryption algorithm, $D$ is the decryption algorithm, and $S$ is the PRNG seeding.

**Theorem 1:**

SSEA3 is computationally secure with two AES-256 encryption algorithms each of 3 rounds.

**Proof of Theorem 1:**

1- SSEA3 uses two AES-256 encryption algorithms $E_1$ and $E_2$ with two different S-Boxes S1 and S2. The used encryption algorithms have 3 rounds.
2- SSEA3 has two keys each of 256 bits which are K1 and K2.
3- SSEA3 uses 16 subkeys for each algorithm at each round which is generated by the AES key schedule.

4- The key schedule of AES generates the seeding S for the RC4 stream cipher algorithm from the encryption keys K1 and K2 where S = K1 XOR K2.
5- S is the seeding for the PRNG to generate the controlling sequence. R is the generated output controlling sequence from the PRNG to control the process of choosing one algorithm to encrypt the data, R chooses one subkey from 16 subkeys at each round and R is the input to the second algorithm.
6- The adversary task is to find the seeding of PRNG, and the two keys of the encryption algorithms.
7- The adversary knows all the information about the used encryption algorithm and RC4 stream cipher algorithm.
8- In the above described scenario, the adversary will be unable to accomplish the task of breaking the ciphertext messages unless the attacker has number of chosen plaintext ciphertext pairs. For one plaintext, the attacker needs to try ($2^{12}X2^{12}X2X2^{128}$) possible combinations to get the right subkeys from the two encryption algorithms and to know which algorithm was used to encrypt the data and to know the encrypted output from RC4. For P number of plaintext blocks, the attacker needs to try (($2^{12}X2^{12}X2X2^{128}$) ^ P). The encryption algorithm in the SSEA3 is operated in ECB mode of operation.
9- At every clock, the PRNG chooses one subkey from 16 subkeys at each round of the block cipher algorithm and chooses one algorithm to encrypt the data while the other algorithm encrypt the output from PRNG then we XOR the outputs from the two encryption algorithms.
10- Every plaintext is encrypted with a group of subkeys out of ($2^{12}$ X $2^{12}$) subkeys groups.
11- The adversary has no advantage to learn anything from the PRNG output since its output is not on the communication channel.
12- The attacker cannot apply known plaintext ciphertext attack because the ciphertext is encrypted.
13- Each time the attacker will choose one combination out of (($2^{12}X2^{12}X2X2^{128}$)^P) possible combinations where P is the number of plaintext blocks.
14- The best chance for the attacker is to guess the seeding S of the PRNG which is 256 bits to start the cryptanalysis to find the right controlling sequence.

Therefore, SSEA3 with two AES-256 encryption algorithms each of 3 rounds is computationally secure.

## 7. SSEA3 Attacks

In this section, we describe the different attacks against SSEA3. SSEA3 is secure against the following attacks:

### 7.1 Attack the PRNG

The adversary has no advantage to learn anything from the PRNG output since its output is not on the communication channel.

### 7.2 Attack the Key Schedule

The attacker cannot apply the related key attack for the SSEA3 because the attacker does not know the plaintext is encrypted with algorithm 1 or algorithm 2. Also, the attacker does not know the plaintext is encrypted with which subkeys group. Related key attack contradicts with the design principal of SSEA3 which states that each plaintext chooses one subkey group from 16 subkeys at each round of the encryption algorithm.

### 7.3 Attack the Encryption Architecture Using Linear and Differential Cryptanalysis

The linear and differential cryptanalysis assumes that the key is fixed for the encryption process and the algorithm is fixed which is not the case for SSEA3. Therefore, the cryptanalyst cannot apply the linear and differential cryptanalysis over SSEA3.

### 7.4 Quantum Computer Attack

The quantum computer can perform cryptanalysis on every ciphertext block using Grover's algorithm where the subkeys are fixed which is not the case for SSEA3 where the subkeys are dynamic and the encryption algorithm is dynamic. Key length is 512 bits to stop Grover's quantum algorithm attack.

### 7.5 Supercomputer Attack

The current fastest supercomputer system is the K computer which is ranked on the TOP500 list as the fastest supercomputer at 8.16 peta FLOPS. It consists of 68,544 SPARC64 VIIIfx CPUs. The system is still under construction and will enter service in November 2012 with 864 cabinets. It currently uses 68,544 2.0GHz 8-core SPARC64 VIIIfx processors for a total of 548,352 cores [**15**].

A supercomputer can perform no more than guessing the sequence of the output controlling sequence of SSEA3. Therefore, supercomputer cannot break the SSEA3.

### 7.6 Attack on Synchronization

If SSEA3 is under miss synchronization attack, the SSEA3 will start with new two keys and therefore, new seeding for PRNG. The transmitter and receiver must initialize with the same seeding and two keys using preamble at the beginning of the secure session.

## 8. Comparison between SSEA3 and Standard AES-256 Block Cipher Algorithm

**Table 1 Comparison between AES-256 and SSEA3**

| No. | Property | AES-256 | SSEA3 |
|---|---|---|---|
| 1 | Speed | Speed of 14 Rounds | Speed of 3Rounds |
| 2 | Security Level | 256 bits Key Length | 512 bits key Length |
| 3 | No. of Algorithms | One | Two |
| 4 | Key Length | 256 bits | 512 bits |
| 5 | Design Size | 14 Rounds for encryption and 14 Rounds for Decryption | 3 Rounds for each algorithm and RC4 Stream Cipher Algorithm and Double Size for Decryption |
| 6 | No. of Rounds | 14 | 3 |
| 7 | RC4 as PRNG | No | Yes |
| 8 | Key Schedule | AES Key Schedule | AES Key Schedule |
| 9 | No. of Subkeys at each round | 1 | 16 |
| 10 | Gain | No | Exponential Gain when increasing number of rounds or number of algorithms or number of subkeys at each round |
| 11 | Complexity | 256 bits | 512 bits and $((2^{12}X2^{12}X2X2^{128}) \wedge P)$ possible combinations, and P is number of Plaintext blocks |
| 12 | Cryptanalysis | Yes | No |
| 13 | Side Channel Attacks | Yes | No |
| 14 | Brute Force Attack | Infeasible for attacker | Infeasible for attacker |
| 15 | Synchronization | Simple | Hard |
| 16 | QC Attacks | No Grover attack | No Grover attack |

## 9. Conclusion and Future Work

In this paper, according to the characteristics of SSEA and its advantages, we could conclude that SSEA family of architectures can resist QC attacks and cryptanalysis attacks. We choose to use SSEA3 as its security level is higher than the security level of SSEA1 and SSEA2 and SSEA1 and SSEA2 have the same speed but SSEA3 has higher speed since it has only 3 rounds. SSEA3 decryption needs double size the encryption design size to allow the encryption speed to be same as the decryption speed.

SSEA security level has exponential gain as the number of rounds increased or the number of subkeys at each round increased or the number of algorithms increased. SSEA is the first encryption algorithm that used the unpredictability principal to add PRNG to the encryption design to hide which algorithm is used to encrypt the data or which subkey is used at each round or to mask the output ciphertext with the encrypted bits stream from RC4 stream cipher algorithm. SSEA is the first encryption algorithm that is immune to cryptanalysis.

The results prove that: the architecture with the advantages of high speed and high security level can be implemented for post quantum cryptography. The SSEA architecture is a strong barrier for cryptanalysis. Besides, each plaintext block is encrypted with a different algorithm and different subkeys group which is an obstacle for cryptanalysis.

In this paper, we proposed a new encryption architecture which is called the spread spectrum encryption architecture. This computationally unbreakable encryption architecture is based on the unpredictability principle where we choose one subkey from 16 subkeys at each round and two algorithms encrypt the plaintext blocks. Our new designed computationally unbreakable SSEA model is easily implemented in both software and hardware. This new encryption architecture will be an essential architecture to the field of post-quantum cryptography. Our future work is to deploy SSEA3 in public key cipher systems on ECC.

## 10. References


[1] Peter W. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, In IEEE Symposium on Foundations of Computer Science, pages 124–134, 1994.

[2] Lov K. Grover, “A Fast Quantum Mechanical Algorithm for Database Search”, Proceedings, STOC 1996, Philadelphia PA, USA, pages 212-219.

[3] Lov Grover, “Quantum Computers can Search Arbitrarily Large Databases by a Single Query”, Phys. Rev., Letter 79, 4709-4712, 1997.

[4] Bennett, Bernstein, Brassard, and Vazirani, “The strengths and weaknesses of quantum computation”, SIAM Journal on Computing 26(5): 1510-1523, 1997.

[5] Steve Babbage, Christophe De Canni`ere, Anne Canteaut, Carlos Cid, Henri Gilbert, Thomas Johansson, Matthew Parker, Bart Preneel, Vincent Rijmen and Matthew Robshaw, “The eSTREAM Portfolio Final Report”, April 15, 2008.

[6] Akihiro Yamamura and Hirokazu Ishizuka, “Quantum Cryptanalysis of Block Ciphers”, Communications Research Laboratory, Nukui-Kitamachi Koganei, Tokyo, Japan, Pages 35-43, 2000.

[7] Gilles Piret and François-Xavier Standaert, “Provable security of block ciphers against linear cryptanalysis: a mission impossible?”, Springer (LNCS 2008) 50:325–338, 2009.

[8] Hamdy S. Soliman and Mohammed Omari, “Application of Synchronous Dynamic Encryption System in Mobile Wireless Domains”, Proceedings of the 1ST ACM international workshop on Quality of service & security in wireless and mobile networks, Montreal, Quebec, Canada, Pages: 24 – 30, 2005.

[9] Bo Dömstedt, and Jesper Jansson, “The Theory of Dynamic Encryption, a New Approach to Cryptography”, Dept. of Computer Science, Lund University, Lund, Sweden, 2000.
[10] Tim S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler, “Breaking Ciphers with COPACOBANA A Cost-Optimized Parallel Code Breaker”. In Cryptographic Hardware and Embedded Systems, CHES 2006, Proceedings of the 8th International Workshop, Yokohama, Japan, LNCS, Springer-Verlag, October 10-13, 2006.
[11] Sandy Harris, “Exploring Cipher space: Combining stream ciphers and block ciphers”, eprint, IACR, November, 2008.
[12] http://csrc.nist.gov/archive/aes/round2/conf3/papers/04-slucks.pdf
[13] National Institute of Standards and Technology. Advanced Encryption Standard (AES). Federal Information Processing Standards Publications FIPS 197 (November 2001) http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[14] Peeter Laud, Semantics and Program Analysis of Computationally Secure Information Flow, Lecture Notes in Computer Science, 2001, Volume 2028, 2001, 77-91.
[15] Mark Hachman, Japan 'K Computer' on Top of TOP500 Supercomputer List, November 14, 2011, PC Magazine.