HOCHSCHULE SRH HEIDELBERG Intelligence in Learning

PSEUDO RANDOM NUMBER GENERATOR FOR CRYPTOGRAPHY

By

Girish Vadlamudi (11013327)

Yashwanth Varakala(11013309)

Sahithi Elaprolu (11013406)

Jitendra Balwada (11013322)

Dept of Information Technology

Under the guidance of

Prof. Mr. Karl Izsak

ABSTRACT:

The major goal of cryptography is to prevent data from being read by any third party. Most transmission systems use a private-key cryptosystem. This system uses a secret key to encrypt and decrypt data which is shared between the sender and receiver. The private keys are distributed and destroyed periodically. Randomness has many uses in science, art, statistics, cryptography, gaming, gambling, and other fields. For example, random assignment in randomized controlled trials helps scientists to test hypotheses, and random numbers or pseudorandom numbers help video games such as video poker. By using Pseudo Random Generator encrypt and decrypt keys are generated and in a high secure state. Pseudo Random Number Generator (PRNG) refers to an algorithm that uses mathematical formulas to produce sequences of random numbers. PRNGs generate a sequence of numbers approximating the properties of random numbers. A PRNG starts from an arbitrary starting state using a seed state. Many numbers are generated in a short time and can also be reproduced later, if the starting point in the sequence is known. Hence, the numbers are deterministic and efficient.

***KEYWORDS:* cryptography, cryptosystem, encrypt, decrypt, Randomness, Pseudo Random number**

MOTIVATION:

Motivation for choosing this research of Pseudo Random Number Generator is to get the high secured encrypt and decrypt keys, designing, testing and carefully integrating, implementing high quality random number generator in cryptographic systems.

INTRODUCTION

RANDOM NUMBERS:

Random numbers are characterized by the fact that their value cannot be predicted. Or, in other words, if one constructs a sequence of random numbers, the probability distribution of the following random numbers must be completely independent of all the other generated numbers.

RANDOM NUMBER GENERATOR:

A random number generator (RNG) is a device that generates a sequence of numbers or symbols that cannot be reasonably predicted better than by a random chance. Random number generators can be true hardware random-number generators (HRNG), which generate genuinely random numbers, or pseudo-random number generators (PRNG), which generate numbers that look random, but are actually deterministic, and can be reproduced if the state of the PRNG is known.

PSEUDO RANDOM NUMBERS:

Approach: Arithmetically generation (calculation) of random numbers

“Pseudo” because generating numbers using a known method removes the potential for true randomness.

GENERATION OF PSEUDO RANDOM NUMBERS:

The generation of pseudo-random numbers is an important and common task in computer programming. While cryptography and certain numerical algorithms require a very high degree of *apparent* randomness, many other operations only need a modest amount of unpredictability. Computational algorithms that can produce long sequences of apparently random results, which are in fact completely determined by a shorter initial value, known as a seed value or key. As a result, the entire seemingly random sequence can be reproduced if the seed value is known. This type of random number generator is often called a pseudorandom number generator.

PROPERTIES OF PSEUDO RANDOM NUMBERS:

• Important properties of good random number routines:

• Fast

• Portable to different computers

• Have sufficiently long cycle

• Replicable

• Verification and debugging

• Use identical stream of random numbers for different systems

• Closely approximate the ideal statistical properties of

• Uniformity

• Independence

There are various steps in cryptography that call for the use of random numbers. Generating a nonce, initialization vector or cryptographic keying materials all require a random number. The security of basic cryptographic elements largely depends on the underlying random number generator (RNG) that was used. An RNG that is suitable for cryptographic usage is called a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG). The strength of a cryptographic system depends heavily on the properties of these CSPRNGs. Depending on how the generated pseudo-random data is applied, a CSPRNG might need to exhibit some (or all) of these properties:

- It appears random
- Its value is unpredictable in advance
- It cannot be reliably reproduced after generation

TEST FOR RANDOM NUMBERS

Two categories:

- Testing for uniformity:

$$H0: R_i \sim U[0,1]$$

$$H1: R_i \nsim U[0,1]$$

- Failure to reject the null hypothesis, H0, means that evidence of nonuniformity has not been detected.
- Testing for independence:

$$H0: R_i \sim \text{independent}$$

$$H1: R_i \nsim \text{independent}$$

- Failure to reject the null hypothesis, H0, means that evidence of dependence has not been detected.
- Level of significance α, the probability of rejecting H0 when it is true: α = P(reject H0 | H0 is true)
- When to use these tests:

➢ If a well-known simulation language or random-number generator is used, it is probably unnecessary to test

➢ If the generator is not explicitly known or documented, e.g., spreadsheet programs, symbolic/numerical calculators, tests should be applied to many sample numbers.

TYPES OF TESTS

- Theoretical tests: evaluate the choices of m, a, and c without generating any numbers
- Empirical tests: applied to actual sequences of numbers produced.
    - Our emphasis.

RANDOM NUMBER DISTRIBUTION:

- In this test we can see the distribution of random numbers on a graph which involves Occurrence v/s Number.
- The theoretical values in this graph is represented with the blue straight line.
- The practical values are represented with the red line and the line should scattered which represents the randomness.

CHI-SQUARE TEST:

- Most commonly used test
- Can be used for any distribution
- Prepare a histogram of the observed data. Compare observed frequencies with theoretical k = Number of cells oi = Observed frequency for I th , cell ei = Expected frequency
- D=0 ⇒ Exact fit

$$\chi^2 = \sum \frac{(f_e - f_o)^2}{f_e}$$

- D has a chi-square distribution with k-1 degrees of freedom. ⇒ Compare D with χ2[1-α; k-1] Pass with confidence α if D is less.

KOLOGMORE-COMPLEXITY TEST:

In algorithmic information theory (a subfield of computer science and mathematics), the Kolmogorov complexity of an object, such as a piece of text, is the length of the shortest computer program that produces the object as output. It is a measure of the computational resources needed to specify the object, and is also known as algorithmic complexity, Solomon off –Kolmogorov–Chaitin complexity, program-size complexity, descriptive complexity, or algorithmic entropy

The notion of Kolmogorov complexity can be used to state and prove impossibility results akin to Cantor's diagonal argument Gödel's incompleteness theorem and Turing's halting problem. In particular for almost all objects, it is not possible to compute even a lower bound for its Kolmogorov complexity, let alone its exact value.

RUN TEST:

The run test (also *runs test*, Wald-Wolfowitz test according to Abraham Wald and Jacob Wolfowitz_iteration test or Geary test**)** is a non-parametric test for the randomness of a sequence. The starting point is an urn model with two types of spheres (dichotomous population). N balls are removed, and the hypothesis is to be tested that the removal was random.

The first step in the runs test is to count the number of runs in the data sequence. There are several ways to define runs in the literature, however, in all cases the formulation must produce a dichotomous sequence of values.

- A few features of a run
    - two characteristics: number of runs and the length of run
    - an up run is a sequence of numbers each of which is succeeded by a larger number; a down run is a sequence of numbers each of which is succeeded by a smaller number

GAP DISTRIBUTION:

- The gap test is used to determine the significance of the interval between recurrence of the same digit.
- A gap of length *x* occurs between the recurrence of some digit.
- There is a total of eighteen 3's in the list. Thus only 17 gaps can occur.
- The probability of a gap length can be determined by a Bernoulli trail.

$$P(\text{gap of } n) = P(x \neq 3)P(x \neq 3)\ldots P(x \neq 3)P(x = 3)$$

- If we are only concerned with digits between 0 and 9, then

$$P(\text{gap of } n) = 0.9^n 0.1 P(\text{gap of } n) = 0.9^n 0.1$$

- The theoretical frequency distribution for randomly ordered digits is given by

$$P(gap \leq x) = F(x) = 0.1 \sum_{n=0}^{x} (0.9)^n = 1 - 0.9^{x+1}$$

RANDOM NUMBER PAIR DISTRIBUTION:

The pair distribution function describes the distribution of distances between pairs of particles contained within a given volume. Mathematically, if *a* and *b* are two particles in a fluid, the pair distribution function of *b* with respect to *a*, denoted by $g_{ab}(r)$ is the probability of finding the *particle b* at distance r *from a*, with *a* taken as the origin of coordinates.

Microsoft Visual Studio:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,[8] C++, C++/CLI, Visual Basic .NET, C#, F#,[9] JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python,[10] Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers".

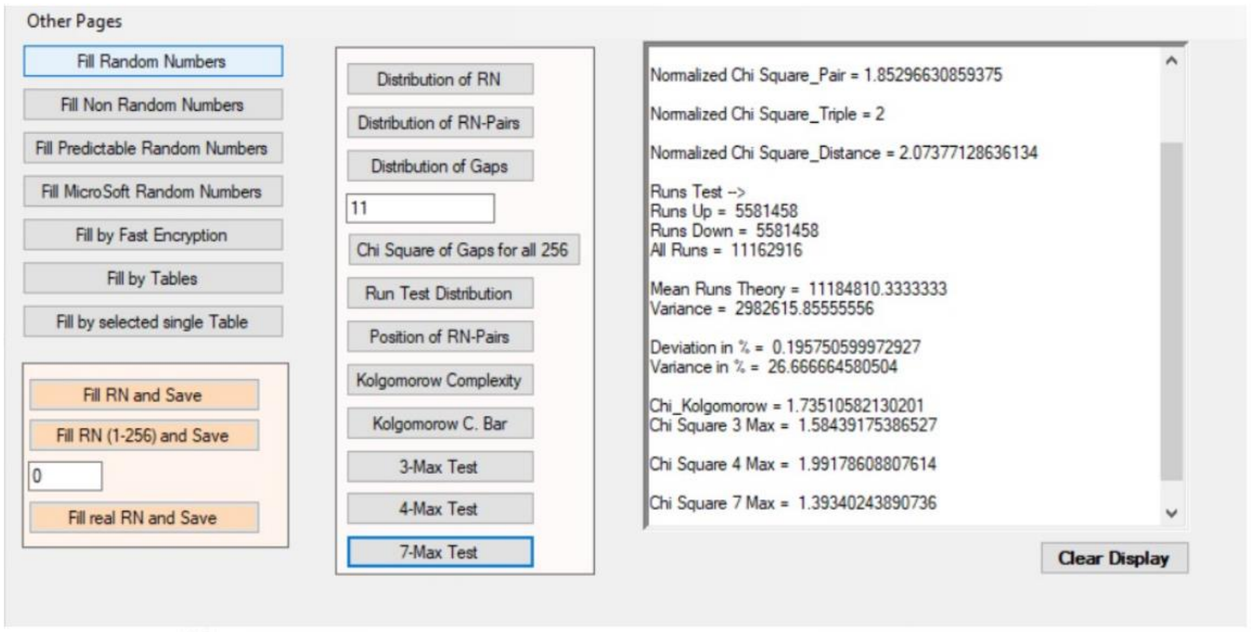
PROGRAMM FOR CONDUCTING TESTS

program.txt

RESULTS

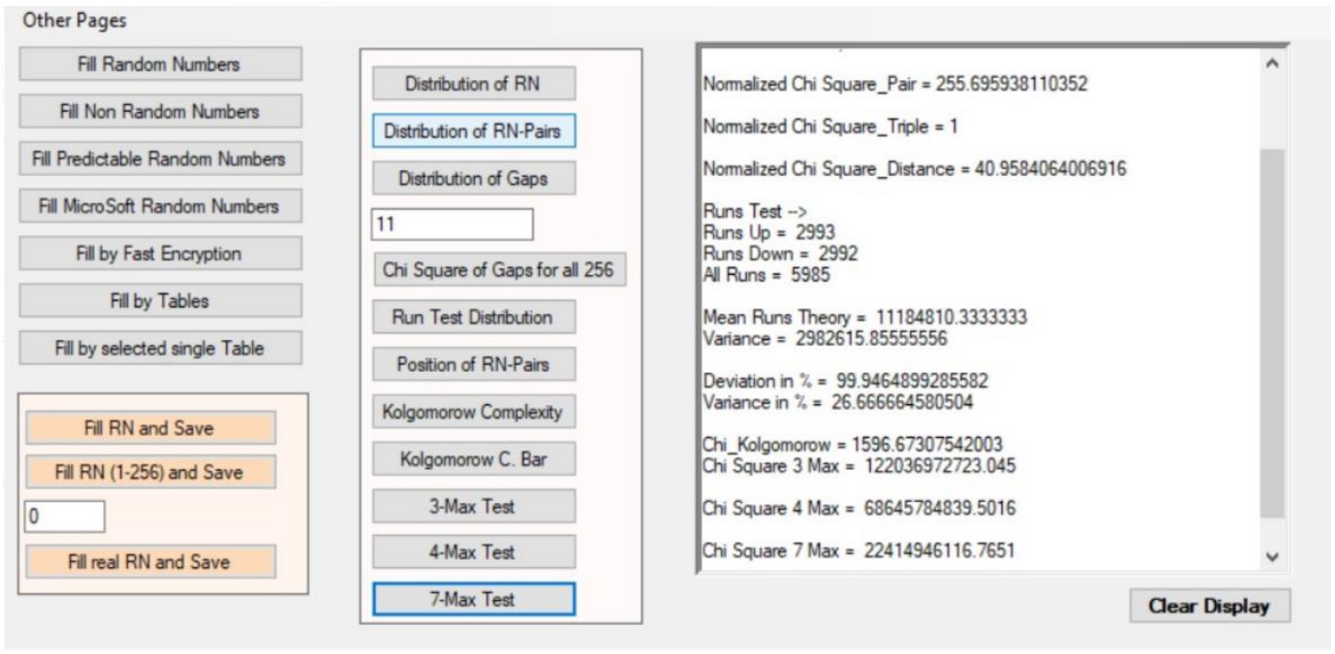
Modified Random numbers result:

Generated Random numbers are divided into half and added to the same sequence



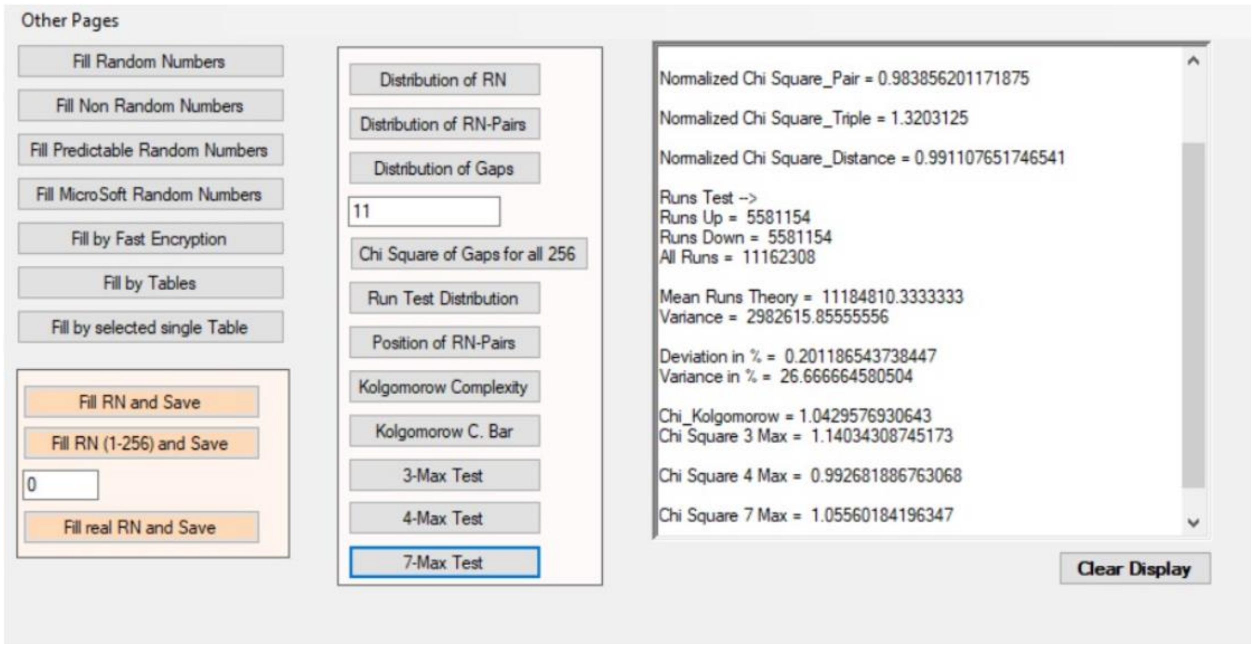
```
Normalized Chi Square_Pair = 1.85296630859375

Normalized Chi Square_Triple = 2

Normalized Chi Square_Distance = 2.07377128636134

Runs Test -->
Runs Up = 5581458
Runs Down = 5581458
All Runs = 11162916

Mean Runs Theory = 11184810.3333333
Variance = 2982615.85555556

Deviation in % = 0.195750599972927
Variance in % = 26.666664580504

Chi_Kolgomorow = 1.73510582130201
Chi Square 3 Max = 1.58439175386527

Chi Square 4 Max = 1.99178608807614

Chi Square 7 Max = 1.39340243890736
```

Generated with the system clock:



```
Normalized Chi Square_Pair = 255.695938110352

Normalized Chi Square_Triple = 1

Normalized Chi Square_Distance = 40.9584064006916

Runs Test -->
Runs Up = 2993
Runs Down = 2992
All Runs = 5985

Mean Runs Theory = 11184810.3333333
Variance = 2982615.85555556

Deviation in % = 99.9464899285582
Variance in % = 26.666664580504

Chi_Kolgomorow = 1596.67307542003
Chi Square 3 Max = 122036972723.045

Chi Square 4 Max = 68645784839.5016

Chi Square 7 Max = 22414946116.7651
```

Generated Random numbers with the c# programm



```
Normalized Chi Square_Pair = 0.983856201171875

Normalized Chi Square_Triple = 1.3203125

Normalized Chi Square_Distance = 0.991107651746541

Runs Test -->
Runs Up =  5581154
Runs Down =  5581154
All Runs =  11162308

Mean Runs Theory =  11184810.3333333
Variance =  2982615.85555556

Deviation in % =  0.201186543738447
Variance in % =  26.666664580504

Chi_Kolgomorow = 1.0429576930643
Chi Square 3 Max =  1.14034308745173

Chi Square 4 Max =  0.992681886763068

Chi Square 7 Max =  1.05560184196347
```

CONCLUSION

For cryptography, the random number generator is very much important and in many aspects like generating the key very securely.

To obtain the Cryptographically secure Pseudorandom number generator the random number generator should pass all tests.

After looking at the test results, we can say that the random numbers generated using **RNGCryptoServiceProvider Class** might be the best method to generate the random numbers

REFERENCES:

https://en.wikipedia.org/wiki/Randomness_tests

https://www.cse.wustl.edu/~jain/cse567-08/ftp/k_27trg.pdf

https://www.eg.bucknell.edu/~xmeng/Course/CS6337/Note/master/node42.html

https://www.digit.in/technology-guides/fasttrack-to-cryptography/uses-of-cryptography.html

en.wikipedia.org/wiki/Applications_of_randomness

https://en.wikipedia.org/wiki/Pseudorandom_number_generator

COMPARISON OF AES vs DES:

DES: It is a symmetric block cipher was introduced by the National Institute of Standard and Technology (NIST) in 1977. It is an implementation of Feistel Structure (a multi-round cipher that divides the whole text into two parts and works on each part individually). It works on 64- bit input key and uses 56- bit shared key to produce the ciphertext of 64-bit. In DES, whole plain text is divided into two parts of 32- a bit each before processing and the same operations are performed on individual parts. Each part undergoes an operation of 16 rounds and after those operations, the final permutation is done to obtain the 64-bit ciphertext.

AES: It is one of the most widely used symmetric block cipher algorithm used nowadays. It was introduced by the National Institute of Standard and Technology in 2001. It is at least six times faster than triple DES. Unlike DES, it works on the principle of 'Substitution and Permutation'. It follows an iterative approach. AES works on bytes rather than bits. In AES, plain text is considered to be 126 bits equivalent to 16 bytes with the secret key of 128 bits which together forms a matrix of 4×4 (having 4 rows and 4 columns). After this step, it performs 10 rounds. Each round has its subprocesses in which 9 rounds include the process of Sub bytes, Shift Rows, Mix Columns and Add Round Keys and the 10th round does include all the above operations excluding 'Mix columns' to produce the 126- bit ciphertext. In AES number of rounds depends on the size of the key, i.e. 10 rounds for 128- bit keys, 12 rounds for 192- bit key and 14 rounds for 256- bit keys. It is used in many protocols like TLS, SSL and various modern applications that require high encryption security. AES is also used for hardware that requires high throughput.

DIFFERENCE:

1. The main difference between DES vs AES is the process of encrypting. In DES, the plaintext is divided into two halves before further processing whereas in AES whole block there is no division and the whole block is processed together to produce the ciphertext.
2. AES is comparatively much faster than DES and is capable of encrypting large files in a fraction of seconds as compared to DES.
3. Because of the small bit size of the shared key used in DES, it is considered to be less secure than AES. DES is considered to more vulnerable to brute-force attacks whereas AES has not been encountered to any serious attacks as of now.
4. Implementation of Algorithm is evaluated on the basis of flexibility and AES is comparatively more flexible than DES as it allows the text of various length including 128, 192, 256 bits whereas DES allows the encryption of text of fixed 64 bits.
5. Functions used in the processing of DES rounds are Expansion, Permutation, and Substitution, XOR operation with round key whereas the functions used in rounds of AES are Sub bytes, Shift Rows, Mix Columns and Add Round Keys.
6. AES is practically efficient with both hardware and software implementations, unlike DES which was initially efficient with only hardware

7. Comparison Table 1

Basis of Comparison between DES vs AES	DES	AES
Developed	DES was developed in 1977	AES was developed in 2001
Full Form	DES stands for Data Encryption Standard	AES stands for Advanced Encryption Standard
Principle	DES follows the principle of Feistel Structure	AES s based on the principle of Substitution and Permutation
Plaintext	Plaintext is of 64 bits	Plaintext can be 128, 192, 256 bits
Ciphertext	Generate Ciphertext of 64 bits	Can Generate Ciphertext of 128, 192, 256 bits
Key Length	Key length is 56 bits	Key length can be 128, 192, 256 bits
Rounds	DES contains a fixed number of rounds, i.e 16	AES contains a variable number of rounds depending on the size of the input, i.e. 10 rounds for 128 bit, 12 rounds for 192 bit and 14 rounds for 256 bits
Security	DES is less secure and hardly used now	AES is much more secure than DES and it widely used nowadays
Speed	DES is comparatively slower than AES	AES is faster than DES

Conclusion

Both DES vs AES is used to encrypt the data and are useful in their own way. AES came as the successor of DES to overcome its drawbacks. AES is also accepted by the U.S. government and has been accepted as a reliable algorithm to secure the classified information. Although DES had made great contributions in the field of data security it is now been replaced by AES in the areas of high security.

Comparison Table 2

AES	DES
AES stands for Advanced Encryption Standard	DES stands for Data Encryption Standard.
AES allows the data length (plain text size) of 128, 192 and 256 bits.	Data encryption standard takes 64-bit plaintext as a input and creates 64-bit Ciphertext i.e. it encrypts data in a block of size 64-bits per block.
AES divide plaintext into 16 bytes (128-bit) blocks and treats each block as a 4×4 State array and supporting three different key lengths, 128, 192. and 256 bits.	In DES plaintext message is divided into size 64-bit block each and encrypted using 56-bit key at the initial level.
The number of rounds are 10, is for the case when the encryption key is 128 bit long. (As mentioned earlier, the number of rounds is 12 when the key is 192 bits and 14 when the key is 256.)	The left plaintext and right plaintext goes through 16 rounds of encryption process along with 16 different keys for each round.
AES was designed by Vincent Rijmen and Joan Daemen.	DES was designed by IBM.
AES is faster.	DES is comparatively slower.
Subbytes, Shiftrows, Mix columns, Addroundkeys.	Expansion Permutation, Xor, S-box, P-box, Xor, and Swap.
10 rounds for 128-bit algorithm 12 rounds for 192-bit algorithm 14 rounds for 256-bit algorithm	Expansion Permutation, Xor, S-box, P-box, Xor, and Swap.
AES has large secret key comparatively hence, more secure.	DES has a smaller key which is less secure.