2006

Differential cryptanalysis of substitution permutation networks and Rijndael-like ciphers

Gnanasekaran Sakthivel

The Role of Diffusion in Rijndael
Project Report

Gnanasekaran Sakthivel
Department of Computer Science
Rochester Institute of Technology

April 8, 2004

Contents

Abstract

Generally, in block ciphers that follow the Substitution Permutation Networks strategy, the round transformation include a non-linear substitution, a linear diffusion and the key addition components. Most of the research has been focused on the design of the substitution in

order to strengthen the cipher against the linear and differential cryptanalysis. But in ciphers similar to Rijndael, the focus is more towards the diffusion component. The goal of this project is to experiment with the linear transformation that is unique to Rijndael in the context of differential analysis. The analysis in this project will be on the reduced round variants of the cipher. The ciphers that we take for our experiment are the Rijndael with no diffusion at all and the Rijndael with a weakened diffusion component. Although this is the final objective of the project, we will begin with and spend a reasonable amount of effort with the implementation and the differential cryptanalysis of the standard substitution permutation networks (SPN). Then the same analysis will be extended to the reduced round variant of the Rijndael.

1 Background

This section gives some important definitions with short descriptions that will help understanding the structure of Rijndael. The important security aspects behind the design is also discussed at the end of each section.

The Rijndael algorithm, a symmetric key cryptosystem and iterated key block cipher system, designed by Vincent Rijmen and Joan Daemen, was selected as the Advanced Encryption Standard (AES) in the year 2000 by the National Institute of Standards Technology (NIST). Most of the AES-competitors were, although secure with respect to several aspects, ruled out for one or more of the reasons [22]. Rijndael was evaluated to be relatively best in all aspects analyzed and expected. Lots of research has been done on the line of cryptanalysis of Rijndael since before the days it was presented as a candidate for AES. As a result of these research, one could say that Rijndael is secure. As Rijndael is well designed taking the known attacks into consideration and shown by many recent publications on the analysis of Rijndael, it is stated by the researchers that Rijndael would survive for a reasonable years to come. The important aspects of this project are

- The implementation and differential analysis of the substitution permutation networks (SPN) cipher
- The SPN implementation will be generalized so that the analysis can be performed efficiently
- The same analysis is extended onto the reduced round variant(s) of Rijndael. There are various possible modifications that can be applied on the reduced round variant. We choose few of them that will be meaningful in our context. A few target ciphers are

- Rijndael without either the shift-rows and/or the mix-columns component
- Rijndael with shift-rows component altered to minimize the diffusion effect
- Rijndael with mix-columns component altered to minimize the diffusion effect
- All the above possibilities will be analyzed also by varying the strength of s-boxes.

All of our analysis will be on the reduced round variants. In the following subsections we introduce the Rijndael algorithm briefly.

1.1 Block Ciphers

In this paper, the term cipher means the corresponding cryptographic algorithm. A block cipher is a set of boolean transformations operating on n_b-bit vectors (which are called blocks) [12]. In other words, a block cipher transforms plaintext blocks of a fixed length n_b-bit to cipher text blocks of the same length under the influence of a cipher key k. Usually the boolean transformation of the block ciphers is divided into three layers namely substitution, diffusion and key mixing. This transformation is key-dependent. The size of the key may determine the number of transformations. In general, the security of this kind of system is achieved by repeatedly applying the transformation. In block ciphers, the input message will be divided into plaintext blocks whose size is equal to the block-size of the cipher. In Data Encryption Standard [DES], the predecessor to AES, the block size is 64 bits. In Rijndael we have 3 variable block sizes namely 128, 192 and 256, whereas AES (as chosen and published by NIST) has restricted the block size only to 128 bits [8].

1.2 Iterative and Iterated Block Ciphers

In iterative block ciphers, the transformations are iterated many times. Each iteration is called a round, and the corresponding transformation is called the round transformation. The round transformations are key-dependent. Each round transformation may or may not be unique and different from the cipher's other round transformations. Each round will have a different key, and the round keys are computed from the cipher key [7]. The algorithm used to derive the individual round keys from the cipher key is called the key schedule algorithm. If the iterative block cipher has same round

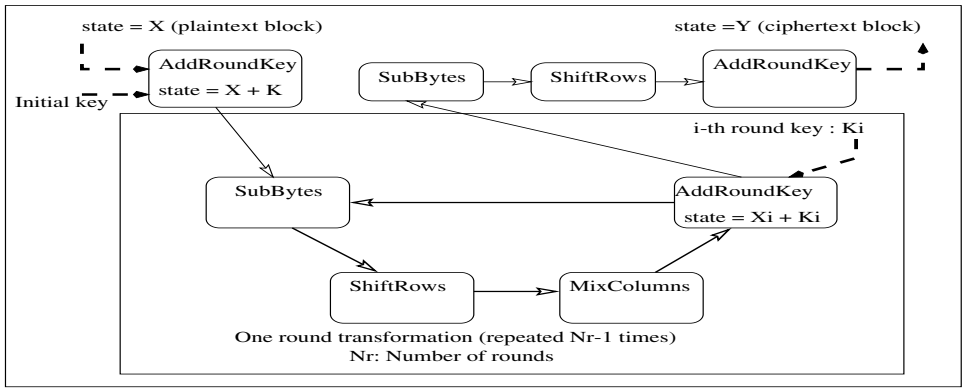

Figure 1: Encryption

transformation for all the rounds (except the first and the last rounds which may be slightly different), then the cipher is called the iterated block cipher [7].

1.3 Key Alternating Ciphers

In this type of ciphers, the round transformation has no involvement of the key. The cipher is considered as an alternated application of the round transformation and the key addition. Usually, the key addition is the simple XOR operation [7].

1.4 Rijndael: A Short Definition

Rijndael belongs to the type of key-alternating block ciphers. It is defined as given below.

$$B[K] = f(K_r) \bullet p(r) \bullet f(K_{r-1}) \bullet p(r-1) \bullet ... \bullet f(K_1) \bullet p(1) \bullet f(K_0),$$

where $f(K_i)$ is the i^th round key addition and the $p(i)$ is the i^th round transformation. We present further details about the round transformation in the next sub section [7].

We will discuss the design of Rijndael tuned to our further insights, that is, we present the security issues behind the design of each component in their respective sections. Above is the diagram that represents the basic structure of the encryption process in Rijndael [Figure 1]. The block size (note that the plain text is divided into blocks), the key size and the number of rounds are suggested as follows. In Rijndael algorithm, the block sizes

can be 128, 192 and 256 bits. The allowable key sizes are 128 bits, 192 bits and 256 bits. And the number of rounds respective to the key sizes are 10, 12 and 14 [21]. We briefly explain what every functional component is for. For the experimental purpose of this project, 128-bit block and 128-bit key size will be used.

2 Rijndael: An introduction

2.1 General terminologies and concepts

State, Plaintext Block and the Ciphertext Block The state is an intermediate result of the cipher. The input is mapped to a state (which is called the initial state) and the output is mapped from the result state. State is viewed as a two dimensional matrix of bytes. The row is fixed to 4 in order to gain good performance in hardware implementations, especially in 32 bit architectures. The number of columns, which is denoted as Nb, varies according to the input block size and will be 4 if the block size is 128 bits(16 bytes), 5 if the block size if 192 bits(20 bytes) and 6 if the block size is 256 bits(24 bytes). The key is also viewed as the state. The plaintext block, which may be 16 or 20 or 24 bytes, is denoted by p_0,p_1,p_2,...,$p_{(4.Nb-1)}$. Similarly the cipher text block is denoted by c_0,c_1,c_2,...,$c_{(4.Nb-1)}$. And the state is denoted by $a_{i,j}$, $0 < i < 4$, $0 < j < Nb$. Where $a_{i,j}$ denotes the byte in row i and column j. The input bytes p_0,p_1,p_2,...,$p_{4.Nb-1}$ are mapped to the state bytes in the order $a_{0,0}$, $a_{1,0}$, $a_{2,0}$, $a_{3,0}$, $a_{0,1}$, $a_{1,1}$, $a_{2,1}$, $a_{3,1}$, ...

Algebraic Nature of Rijndael The algebraic nature of Rijndael is of finite fields [16, 3]. The components of Rijndael can also be defined in the field GF(2), while this is not true in the case of other block ciphers. In Rijndael, the bytes of the input are considered as polynomials in the field GF(2^8). There are two operations that are basically necessary for the algorithm, addition and multiplication of polynomials. Addition of polynomials is simply the addition of corresponding coefficients modulo 2 [27, 15]. Multiplication of polynomials has to be reduced to a polynomial of degree lesser than 8, for it is in GF(2^8). The irreducible polynomial chosen for this purpose is x^8 + x^4 + x^3 + x + 1 [7]. With this irreducible polynomial, the representation for the field GF(2^8) is constructed. Examples for polynomial addition and multiplication are given below. The coefficients of the polynomials in GF(2^8) can be either 0 or 1.

Addition

$$(x^7 + x^5 + x^4 + x + 1) + (x^6 + x^5 + x^4 + x^3 + 1) = (x^7 + x^3 + x)$$

Multiplication

$$(x^5 + x^3 + 1) * (x^6 + x^4 + 1) = (x^{11} + x^9 + x^5 + x^9 + x^7 + x^3 + x^6 + x^4 + 1)$$
$$= (x^{11} + x^7 + x^6 + x^5 + x^4 + 1) \pmod{x^8 + x^4 + x^3 + x + 1} = (x^5 + x^3 + 1)$$

Security margin of a cipher and other security measures For a cipher specified with n rounds, if there exists a successful cryptanalytic attack (successful in the sense, the attack algorithm should be faster than the exhaustive key search) against a reduced-round version of the cipher with k rounds, the cipher then is said to have an absolute security margin of n-k rounds or a relative security margin of $\frac{(n-k)}{n}$ [7]. There are other quantitative measures that may also be taken to determine the strength of the cipher like finding lower bounds on th respect to the complexity of specific attacks, finding the maximum input-output correlation in the context of the linear cryptanalysis and etc.[7, 23].

2.2 Components of Rijndael

AddRoundKey $[S' = S + K]$ This component takes a state as the input and XORs it with the Key. All the rounds of Rijndael can use the same key or different keys. The decision whether to use same key or different keys for all the rounds is discussed in the reference [7]. This AddRoundKey operation can be represented as the addition of matrices.

SubBytes This component is similar to the substitution in Substitution Permutation Networks (SPN) cipher in the sense that it is substitution [12]. The substitution in SPN has no linear or algebraic definition, whereas the substitution in Rijndael is defined algebraically [7]. Linearity is there if and only if $f(A + B) = f(A) + f(B)$. Anyone can easily verify that the other components are linear in nature The nonlinearity property is proved to be a strong cryptographic primitive [21]. The other desirable properties stated by the authors in the paper [12] are invertibility, minimization of the largest non-trivial correlation between linear combinations of input bits and linear combination of output bits, minimization of the largest non-trivial value in the EX-OR table, complexity of its algebraic expression in GF(2^8) and simplicity of description. Most of these criteria are now necessary for

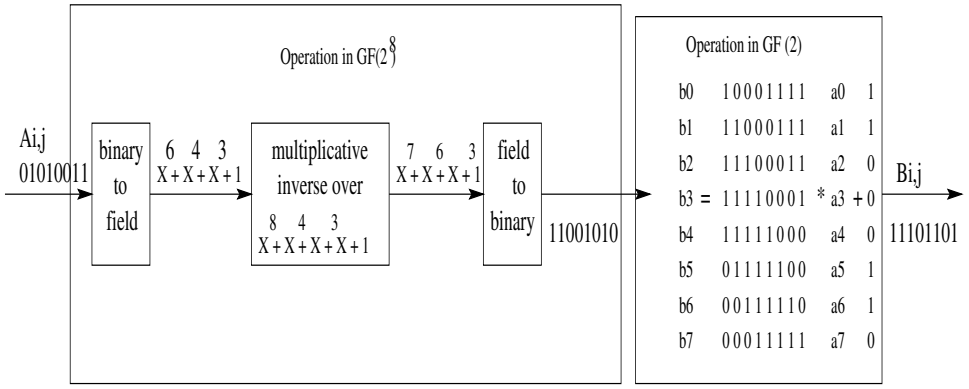

Figure 2: Substitution

any block cipher to be secure against the attacks that are already known, especially the linear and differential cryptanalysis.

The functionality of S-Box includes operations in GF(2^8) and operations in GF(2). [GF: Galois or Finite Field]4,2]. Note that the S-Box takes a byte as input and it has two sub components. The first one is finding the multiplicative inverse of the field representation of the input byte. As it is a simple algebraic expression by itself, it is possible to mount algebraic attacks like interpolation attack, so it is followed by an affine transformation, properly chosen in order to make the SubBytes a complex algebraic expression, but preserving the non linearity property as a whole. The functionality of the s-box of Rijndael is given in Figure 2. The diagram is illustrated with an example.

As the legendary attacks due to linear and differential cryptanalysis exploit the input-output correlation and the propagation of the differences (which is the XOR of selected bits of the input and output) to extract the partial or whole bits of the keys respectively, in order to secure the cipher against the attacks related to these cryptanalysis, the non linearity of the S-Box should satisfy few properties. They are, the maximum input output correlation and the difference propagation probability should be minimum to the extent possible. The substitution component of Rijndael is having these properties at the maximum optimum level. Also the SubBytes component can be implemented as a table look-up operation to gain speed and to preclude timing and differential power attacks. Decryption can be achieved by reversing the steps of the algorithm with few exceptions and with inverted components. As this project concentrate mainly in diffusion characteristics, this component has no significant role other than it is an important

component worth mentioning for it is enforcing the non-linearity, a required primitive, and for the understanding of the whole cipher. There is no specific issues to be discussed for the 128 bit block and key sizes in this component.

ShiftRows [Permutation of bytes at row level] This function is a simple permutation or it is sometimes called transposition. The main function of this layer is to diffuse (spread) the changes made at the SubBytes layer. The diffusion should be well optimized to give resistance against the linear and differential cryptanalysis. The ShiftRows layer operates on row level of the state. The permutation is achieved here by shifting the rows cyclically over different offsets. The four rows should have different offsets. And the shift offsets for various block lengths are given in the reference [8]. The choice of the shift offsets matters in case of the algorithm's strength against differential and saturation attacks. So in Rijndael, the simplest and the strong options are been chosen for the offsets. For the target block length of 128 bytes, the first row is not shifted, but the second by one left shift, third by two and fourth row by three left shifts.

In this project we will be changing the offsets in order to easily perform the differential analysis and will step into finding the difference (in the strength level) between various optimal offset options.

MixColumns [Permutation at column level] This is an important component worth attention. The sub-bytes component, we discussed before, is actually changing the bytes and this local byte change is diffused (spread) further with the help of the mix-columns operation. This diffusion is one of the three cryptographic primitives desired [16]. As we already discussed, this mix-columns layer and the shift-rows layer together contributes to the diffusion. This permutation is a complete algebraic one as what we see.

Each column of the input state is considered as a polynomial over GF(2^8) and it is multiplied modulo $X^4 + 1$ with a fixed polynomial C(X)

$$C(X) =' 03' x^3 +' 01' x^2 +' 01' x +' 02'$$

Although the polynomial $X^4 + 1$ is not irreducible, Rijndael chooses the polynomial C(X) to be co-prime to $X^4 + 1$, and so the operation is invertible inorder to perform decryption [28]. Another interesting fact is the coefficients of both the polynomial C(x) and the polynomial of a column of the state are by itself polynomials in GF (2^8). So if we take the 0^{th} column of the state, it is

$$a_{30}x_3 + a_{20}x_2 + a_{10}x + a_{00}$$

It is now a polynomial in GF(2^8). In the pursuit of mixing columns, it is multiplied by the polynomial C[x] modulo $x_4 + 1$ to get the result

$$b_{30}x_3 + b_{20}x_2 + b_{10}x + b_{00}$$

Note that the coefficients of the polynomials are themselves polynomials in GF(2^8) , as we discussed earlier. For example, the coefficient '03' in C(x) is actually the hexadecimal representation of a byte, and is nothing but the polynomial x+1.

The multiplication by C(x) modulo $x^4 + 1$ can also be represented as a matrix transformation. For example if we take the 0^{th} column, which has four bytes (a_{00}, a_{10}, a_{20}, a_{30}), the MixColumns layer does the following to get the the 0^{th} column of the result ($b_{00}, b_{10}, b_{20}, b_{30}$). Similarly the operation is repeated for other columns.

$$\begin{pmatrix} b_{00} \\ b_{10} \\ b_{20} \\ b_{30} \end{pmatrix} = \begin{pmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{pmatrix} * \begin{pmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \end{pmatrix}$$

The choice of the coefficients has two advantages, they are simple like 03, 02 and 01 in order to gain easy implementation in hardware (Since multiplication by '02' or by x is just the multiplication by polynomial x, which is a left shift modulo the (Rijndael) polynomial and multiplication by 01 has no processing at all) and also they are chosen to give optimal diffusion in order to preclude differential and linear cryptanalysis [7]. In this layer also, care was and should be taken to fix the cycles per execution in the implementation to avoid timing and power attacks. Because in ordinary implementation, if any of the shift operation involved (in order to multiply by '02') leads to overflow and additional operation, then the cycle time will vary [24]. But there are many implementation methods suggested and can be followed to avoid these attacks. In this project we will carry out the differential analysis either by modifying the constants or by completely removing this layer in order to minimize the optimality of the diffusion.

Key Scheduling Every round has a different key, where each round key is derived from the cipher's original key. The key schedule algorithm has two

subcomponents. First one is the key expansion which is used to derive the expanded key from the cipher key. The expanded key is the concatenation of the individual round keys. The second subcomponent is the round key selection, where in simple to sophisticated ways to select the round keys can be accomplished. But Rijndael has the simple key selection procedure. Nonlinearity, diffusion and symmetry elimination are achieved to avoid some specific kind of attacks [7]. Nonlinearity is achieved by using the same subbytes component, and diffusion is used to efficiently spread the cipher key differences into the expanded key and the symmetry elimination is achieved by using a different constant in generating each round key.

2.3 Other important aspects

Following are some other aspects that are important in the context of the design of Rijndael. The following is a brief summary of the discussion by designers in their book [7].

Simplicity and Symmetry Simplicity is maintained in the design of every component of the algorithm, which is one of the characteristic imposed by the NIST and expected by everyone from the level of the users to cryptanalysts. Rijndael designers have considered the simplicity in two aspects. One is the specification simplicity and the other one is analysis simplicity. Simplicity in specification or analysis should not be misunderstood by concluding that the simplicity would contribute to some weakness. It is rather working in the opposite way in the case of Rijndael. In the absence of successful cryptanalysis, as everyone would understand out of its simplicity, it will appeal to the credibility of the algorithm. As the simplicity in analysis offers, Rijndael has the ability to demonstrate the way in which the Cipher is strong against known and possible attacks. Although simplicity can be achieved in many ways, Rijndael has achieved it by choosing the symmetry in its components and the choice of their operations. The algorithm has lot of symmetry which is one of the measure taken to achieve the simplicity. Symmetry means each bits of the plaintext is treated similarly in the algorithm. Symmetry is accomplished across the rounds, within the round transformation and in the steps of the round. There is an attack, which is called Slide Attack [1], that exploits the symmetry of any block cipher. The weakness of this attack is: it demands significant symmetry even in the key schedule of the (Iterative) block ciphers[7]. In Rijndael, round constants are used in getting the key for each round. This is one of the simplest ways to eliminate the symmetry in the key schedule. There exists a good bundle

alignment (where a bundle is a group of bits that are together processed as a unit in any component of the layer) to achieve the symmetry. This alignment property is exploited by the so called saturation (or square) attack, but fortunately because of its high complexity, it is feasible efficiently only till six rounds.

Because of the symmetry and alignment properties, it was easy to prove the lower bounds for the complexity of any attacks and to find the maximum difference propagation probability and the maximum input-output correlation, which is not the case in other ciphers which have sophisticated and complicated round transformation [7].

Global and Local optimizations Local optimization concerns the optimization and security measures taken in the round transformation, rather the global optimization is about getting good properties like diffusion at the overall cipher level, in fact the latter deals with the sequence of rounds rather than a single round. Local optimization concentrates on a round transformation, where to obtain non-linearity and diffusion at the local level, the maximum input-output correlation is minimized to the extent possible in the context of linear cryptanalysis and the maximum difference propagation probability is minimized in the context of differential cryptanalysis. Most of the available block ciphers are designed with good local optimization. Specific and unique in the design of Rijndael is the designers have given significant consideration to the global optimization, in this regard, Wide Trail Strategy is an almost unique and an efficient approach taken [7]. The important benefits of targeting the global optimization are cheap non-linear boolean transformations such as small S-Boxes [7]. Also it favors to specify not what the block ciphers should satisfy, but give concrete criteria for the design of components of the round transformation, where the criteria mostly are having diffusion and non-linear components.

2.4 Differential Cryptanalysis

General information Differential cryptanalysis, along with the linear analysis [27, 12], was the well-known chosen-plaintext attack [16] and a successful analysis against the block ciphers. There are two characteristics that the differential analysis exploits from the block ciphers. They are the lack of optimal non-linearity and lack of good diffusion. The main technique behind this analysis is finding the s-boxes which are active. An active s-box is the one that has one or more high probable differentials. The two important terms used in this analysis are the differential and the difference [12]. The

difference is the exclusive-or operation of the bits (either input or output bits). Let us denote the output difference, which is the exclusive-or of the selected output bits of the chosen s-box, as D_Y. And the input difference which is denoted as D_X. Then the differential is a pair (D_X, D_Y), such that the probability that this D_Y occurs given this D_X is much greater than the $1/2^n$, where n is the number of bits (of the input or the block size in other words) involved[12].

In the analyses of a cipher with many rounds, we construct a differential (D_X,D_Y) involving plaintext bits and the bits of the input to the last round of the cipher. We denote the i-th round differential as (D_{Xi}, D_{yi}]. We construct the differentials of the whole cipher after constructing the sequence of high probable differentials from the active s-boxes. We construct the sequence as (D_{X1},D_{Y1}), (D_{X2},D_{Y2}), ..., (D_{Xn},D_{Yn}), such that the permuted D_{yi} is equal to D_{Xi+1} and it should be such that every differential pair should have almost high probability. The key bits get cancelled and so it wouldnot come in the differential expression [12]. The whole sequence is just the concatenation of appropriate differential pairs of s-boxes across the rounds. With this overall differential, it is highly possible to recover a subset of the subkey bits following the last round [27].

Linear transformation It is to be noted that although the non-linear substitution (confusion) layer contributes to the strength of the cipher against the differential analysis, the linear transformation of the algorithm, the permutation, increases the resistance to differential analysis [13]. From the research done on the design on S-boxes relative to the known attacks, it was found that large S-boxes should be chosen, which will take lot of resources in the implementation. So in the design of Rijndael and other related cipher which follow the wide-trail strategy, large resources are rather spent in the linear transformation to provide high multiple- round diffusion [7].

Rijndael by itself is well designed against these analysis, and it is proved strong against this analyses because of its good non-linearity and diffusion optimized both locally and globally. In the pursuit of finding the role of the diffusion in Rijndael, this project includes minimizing the diffusion by the techniques that we already discussed in the corresponding sections and applying this analysis. he reasons we focus mainly on diffusion is due to the fact that in the design of Rijndael, the strength against the differential analysis is unusually gained more through the linear step rather than the non-linear step.

3 Project description and its outcomes

As we discussed earlier at the end of the sections related to shift-rows and the mix-columns components, this project is to analyse the diffusion layer, and may lead to the following conclusions.

3.1 Goals

- Build the standard substitution permutation network [SPN] according to the tutorial by professor Hayes [12]
- Perform the differential analysis to understand the pros and cons of s-boxes and the diffusion in this context
- Extend the above analysis to the reduced round variants of Rijndael
- Describe the importance of both the diffusion components (separately and also as together) through some quantitative results that we get from our analysis
- Discuss about the possible weak choices of the offsets in the shift-rows component and the coefficients of the polynomial C(x), which are the important entities of this diffusion layer, that determine the strength of the Cipher

3.2 Main discussion of the project

There are lot of previous research done along this line and the authors have defined how to construct an efficient key-alternating structure with optimal diffusion [7]. One of the main aims of this project is to experiment this. In this subsection, we briefly discussed the works involved in the project. The project will also include and discuss other interesting aspects that might arise as a result of the mainstream. To achieve this, the project has several steps.

- First of all, to get a good understanding of the differential analysis and possibly its variations, this project includes an implementation of the simple block cipher Substitution Permutation Network (SPN), which is extensively used for academic research. In fact, the purpose of this implementation of SPN is to carry out differential analysis on the SPN cipher according to the references [27, 12]. The implementation will include a program which will do this and the same can be used for learning the practical aspects of the simple differential analysis.

- Setting up the implementation of (ANSI-C) Rijndael provided by the National Institute of Standards Institute. This implementation will be studied well enough to modify the number of rounds and to fix the block and key size to 128-bits. The number of rounds for our analysis will be around 4 to 6. Because the best known attacks are succeeded till 7 rounds.

- Analyse with respect to the shift-rows layer in the context of both the differential analyses and its variations possibly. Here we may pursue to change the offsets used in the shift-rows component and make these analyses possible. The cipher we use here will be the one without the mix-columns layer. So we use the shift-rows component as the only contribution to the diffusion.

- Analyse the mix-columns layer in detail enough to figure out the possibilities of introducing (or finding) weakness into it either by altering the layer to reduce the diffusion or by removing the layer. Here we may attempt to choose different coeffiecients of C(x) or a different polynomial C(x) to make the differential analysis feasible. It is a fact that even though the round transformation used provides low non-linearity or diffusion, repeating the rounds often enough will result in a block cipher that is unbreakable against the linear and differential cryptanalysis [7]. So we will try to carry out the analysis in a reduced round altered version of the cipher. This will help to determine the strength (security margin) of the altered cipher quantitatively with respect to the changes that we make.

- Analyse a cipher that has both the components in it and with knowledge that we gain from the previous analyses.

- As there is a significant consideration given to the linear transformation with respect to the Wide trail strategy, our analysis may also involve setting the s-boxes weaker and do the possible analyses that are above mentioned.

4 Projected Schedule

This project is expected to be completed in a duration of two quarters. The tentative schedule of the project is presented below.

Sep 1-7, SPN Implementation : SPN as given in the tutorial by Professor Howard M Heys or by Prof. Stinson will be implemented

Sep 8-27, Differential analysis: The differential analysis is further studied and applied on the SPN already implemented, and a report will be prepared based on this

Sep 28 to Oct 25, Study of diffusion components: This step involves the study of mathematics behind the Wide trail strategy and the role of linear diffusion layer in this wide trail strategy

Oct 26 to Nov 8, Implementation: As we discussed earlier, there are lot of possibilites for the target cipher. We finalise the possibilites and implement each of them for analysis.

Nov 23 to Dec 20, Carrying out the analysis: This is part I of the analysis, and if we encounter any significant changes to what we planned, we update our further analysis accordingly

Dec 28 to Jan 24, carrying out further analysis: This is part II of the analysis

Jan 25 to Feb 21, Document and report preparation: This step of the project may include final analysis stages besides the main report preparation

Feb 23 to Feb 29, Defense: The project will be defended during this week

5 SPN and Differential Analysis

References

[1] Alex Biryukov, David Wagner, *Slide Attacks*, EUROCRYPT 2000, International Conference on Theory and Application of Cryptograhic Techniques, March 2000 proceedings

[2] Lawrence Brown, Jennifer Seberry, *On the design of permutation P in DES type cryptosystems*

[3] Jung Hee Cheon, Munju Kim, Kwangjo Kim,Jung-Yeun Lee, and Sung-Woo Kang, *Improved Impossible Differential Cryptanalysis of Rijndael and Crypton*, Springer series

[4] Nicolas T Courtois and Josef Piepryzk, *Cryptanalysis of block ciphers with overdefined systems of equations*, IACR eprint server, 2002. Also available at http://eprint.iacr.org/2002/044

[5] Cryptography by Nicolas T Courtois, www.minrank.org/aes

[6] AES page by Evan Damgaar, www.cryptomathic.com/company/aes.html

[7] Joan Daemen and Vincent Rijmen, *The Design of Rijndael. AES - The Advanced Encryption Standard*, Springer

[8] Joan Daemen, Vincent Rijmen, *AES Proposal*, (2000)

[9] Joan Daemen, Vincent Rijmen, *Answers to new observations on Rijndael*, August 11, 2000

[10] Neils Ferguson, John Kesley, Stefan Lucks, Bruce Schneir, M.Stay, D.Wagner, David Wagner, and Doug Whiting, *Improved Cryptanalysis of Rijndael*, Proceedings of Fast Software Encryption - FSE'00, number 1978 in Lecture notes in Computer Science. Pages 213-230. Springer-Verlag, 2000

[11] Niels Ferguson, Richard Schroeppel, and Doug Whiting, *A Simple algebraic representation of Rijndael*, Proceedings of Selected areas in Cryptography - SAC'01, number 2259 in Lecture Notes in Computer Science, pages 103-111. Springer-Verlag, 2001

[12] Howard M Heys, *A Tutorial on Linear and differential Cryptanalysis*

[13] Howard M Heys, *Substitution-Permutation Networks resistant to differential and linear cryptanalysis*, Journal of cryptology (1996) 9:1-19

[14] Thomas Jakobsen and Lars R Knudsen, *The Interpolation Attack on Block Ciphers*

[15] Rudolf Lidl and Harald Neiederreiter, *Introduction to Finite Fields and their Applications*, Cambridge University Press, 1986

[16] Alfred J Menezes, Paul C Van Oorschot and Scott A Vanstone, *Hanbook of Applied Cryptograhy* CRC press, Fifth print, Augus 2001

[17] Sean Murphy and M J B Robshaw, *Essential Algebraic Structure Within the AES*, Information security Group, University of London, UK

[18] Sean Murphy and Matt Robshaw, *New Observations on Rijndael*, Information Security Group, University of London, UK

[19] Modes of AES: www.csrc.nist.gov/CryptoToolkit/modes/proposedmodes

[20] Elisabeth Oswald, Joan Daemen and Vincent Rijmen, *AES - The State of the Art of Rijndael's Security*, October 30, 2002

[21] Bart Preneel, Vincent Rijmen, and Antoon Bosselaers, *Recent Developments in the Design of Conventional Cryptographic Algorithms*, 18 September 1998

[22] Dhiren R Patel, *The AES winner*, AES third conference

[23] Sangwoo Park, Soo Hak Sung, Seongtaek Chee, E-Joong yoon, and Jongin Lim, *on the security of Rindael-Like structures against differential and linear cryptanalysis*, Advances in Cryptology, 2002

[24] Jean-Jacques Quisquater and Francois Koeune, *A timing attack against Rijndael*, June 10, 1999

[25] Rijndael specific, www.rijndael.com

[26] Rijndael designer's page, www.esat.kuleuven.ac.be/rijmen/rijndael.

[27] Douglas R Stinson, *Cryptography Theory and Practice* 2nd edition, CRC Press

[28] Michael Welschenbach, *Cryptography in C and C++*, Apress.