Abstract

A group is said to be *pseudo-free* if an adversary who is allowed to see group description, is unable to solve non-trivial equations. Non-trivial equations are the equations that have no solution in a free group. The notion of pseudo-freeness was introduced by Rivest, who left as a open problems whether such groups exists in cryptography or not. Rivest's conjecture that the RSA group is pseudo-free has been settled by Micciancio for case of static adversary who is given only the group description and his goal is to come up with an equation and solution.

Later on the notion of pseudo-freeness has been extended to adaptive adversary for case of univariate equations by Dario.C, Dario.F and warinschi.B. In case of adaptive pseudo-free groups, the adversary learns solutions of some non-trivial equations before producing a new equation.

In this project we make extension to the notion of pseudo-free groups from univariate to multivariate equations and system of equations in face of adaptive adversary. In our setting along with the group description the adversary is allowed to see some non-trivial equations with their solutions and finally his goal is to come up with new non-trivial equation and solution.

First of all we show triviality for case of adaptive pseudo-free groups[page 30] and then craft definitions with respect to multivariate equations[page 32] and system of equations[page 42]. Therefore in this project we provide

- Formal definition of adaptive pseudo-free groups with multivariate equations and system of equations
- Proof that the "The RSA group is adaptive pseudo-free with respect to multivariate equations"[page 32].

Acknowledgements

I would like to thank

- My Project Supervisor, Dr Bogdan Warinschi for introducing me to this fascinating area of Cryptography and for his guidance throughout the project.
- My Project coordinator, Dr Oliver Ray for his help and guidance throughout the Masters Course, and particularly for his help and advice on my dissertation.
- My family for their constant support and encouragement.

Contents

1 Introduction

Most of the modern cryptography relies on difficulty of solving non trivial equations over computational finite groups. Cryptography speculates pseudo-free groups as a very strong and interesting notion. The notion of pseudo-free groups was first introduced by Hohenberger[18]. She presented an informal definition of such groups. She used them to identify transitive signature schemes along with their variants and found that the adversary cannot efficiently compute their inversion[18]. After Hohenberger, Rivest[31] polished this notion and extended it by defining it formally. He considered pseudo-free groups to be a strong notion and assumed that many other computational assumptions[26] typically used in cryptography, like the hardness of computing discrete logarithms, the RSA assumption and the strong RSA assumption holds the property of pseudo-free groups. Subsequently D.Miccianico[26] proved the pseudo-freeness of the RSA group under the strong RSA assumption for static adversary. He proved pseudo-freeness of RSA group (Z_N^*) at least when $N = PQ$ and is the product of two safe primes such that $P = 2p + 1$ and $Q = 2q + 1$. Here P, Q are safe (odd) primes and p, q are known as Sophie German primes. Safe prime is a special type of prime numbers used in cryptography. In his static setting[26], the adversary is allowed to see the description of the group and finally his goal is to come up with some equation and solution to that equation.

Informally, a computational group is considered to be *pseudo-free* if an adversary who is allowed to see description of the group cannot efficiently compute solutions for non-trivial equations, where as non-trivial equation means the equation whose solution does not exist in a free group. We give the formal description of pseudo-free groups in section[3.4]. Free groups[26] are used in Dolev-Yao model for symbolic analysis of public key cryptographic protocols. From past few years, a lot of study[26] have been carried out to cover the gap between the symbolic Dolev- Yao model and standard computational model used in cryptography in order to achieve computational soundness results for symbolic analysis methods. An important motivation to study pseudo-free groups is to see whether it can be used to extend Dolev-Yao security model with cryptographic functions that make use of such groups in cryptography and lead to many new applications. Other reasons for studying pseudo-free groups are as follows [31]:

- Proofs perhaps might be made easier with the use of stronger assumptions that in turn incorporates many other common cryptographic assumptions like the hardness of computing discrete logarithms and the strong RSA assumption

- If we assume that a group is pseudo-free then it may possibly allow an even wider range of applications, as the strong RSA assumption has been very serviceable in the construction of many cryptographic functions which are not acknowledged to be secure under the standard version of the RSA assumption

- There is no prevailing solution for specific cryptographic primitives, like directed transitive signature schemes whose construction has been linked to Pseudo-freeness

In cryptography[9], Pseudo-free groups so far do not have so many applications and the reason for this is that in most of the cases of RSA group, not only group's description is provided to adversary but also he is given some equations and solutions. Thus after learning solutions to some equations his goal is to output a new equation along with its solution. For example as defined in [9] such cases holds true for signature schemes based on RSA where adversary can output signature to solve equation whose solution does not exist in a free group. Here the adversary can gain access to an oracle through chosen message attack and generates forgery as a solution to new non trivial equation. This problem was noticed by Rivest[31] who left the notion of pseudo-freeness as an open problem and also mentioned, whether pseudo-free groups exist in cryptography?.

1.1 Aims and Objectives

The main aim of the project is to explore the notion of pseudo-free groups with multivariate equations and system of equations to adaptive adversary. Pseudo-free groups so far has been defined for univariate and multivariate equations for static adversary and univariate equations for adaptive adversary. In this project we extend this study and identify pseudo-freeness with respect to an adversary who may learn solutions to other non-trivial equations before solving a new non-trivial equation.

This project has follwing objectives:

- To define adaptive pseudo-freeness with respect to multivariate equations and system of equations.
- To provide constructions that will show how RSA group satisfies our definitions.

The current definitions of pseudo-free groups are not sufficient to make use of such groups in cryptography[26]. The reason is that an adversary[9] that interacts with some cryptographic primitive built on top of a group may obtain additional information not available to him in the game that defines pseudo-freeness. Therefore we explore this notion with the intuition of getting much stronger version of pseudo-free groups..

Adaptive pseudo-freeness[9]: Informally in adaptive pseudo-free groups, the adversary along with the group description is allowed to see some equations (that have no solution in a free group) together with its solutions. Finally aim of adversary is to come up with new non-trivial equation and solution.
In this project we explore the notion of adaptive pseudo-free groups and prove that the RSA is adaptive pseudo-free. Our results in [5.5] holds for multivariate equations but can easily extend to system of equations. The definition of pseudo-free groups in

adaptive setting consists of two design decisions. The first issue with the design is to set equations for which an adversary can see solutions and the second issue is to define what do we actually mean by non-trivial multivariate equations in adaptive setting of pseudo-free groups. We first define triviality[5.4,6.1] of equations and then craft a definition of adaptive pseudo-free groups for case of multivariate equations[Definition 11] and system of equations [Definition 13]. We detail our results next.

1.2 Structure

This report consist of six main sections. The first section is devoted to introduce the basic concept of pseudo-free groups with the motivation of why there is a need to study such groups. Section two presents some preliminaries and mathematical background which is necessary to understand the notion of pseudo-freeness. Section three describes the formal definition of free groups and pseudo-free groups with the implication of such groups to other strong cryptographic assumptions. Section four presents the formal definition of static pseudo-free groups and shows that how RSA group satisfies this definition in case of static adversary. Section five presents adaptive pseudo-free groups and their applications. In this section we introduce our concept of adaptive pseudo-free groups with multivariate equations. First of all we show triviality in case of multivariate equations and then craft definition out of them. Later in this section we prove that the RSA is adaptive pseudo-free with multivariate equations. In Section six we extend the notion of adaptive pseudo-free groups from multivariate to system of equations. Finally in last two sections we discuss some open problems that have been investigated in the notion of pseudo-freeness and as a future scope we show an idea of extracting a more efficient application out of these groups.

2 Preliminaries

2.1 *The RSA Group*[31]- The security of the RSA encryption scheme depends on RSA assumption which works with multiplicative group Z_N^* such that N is the product of two large primes.

RSA assumption-
This assumption states that it is hard for an PPT adversary given an interger *'N'*, e such that e>1 and a random y selected from Z_N^* to find x that satisfies

$$x^e = y(modN)$$

with non-negligible probability.
Where *'N'* is the product of two randomly chosen large primes.

Strong RSA assumption-
This assumption states that it is hard for an PPT adversary given an interger *'N'*, and a random y selected from Z_N^* to find x and an integer e>1 that satisfies

$$x^e = y(modN)$$

with non-negligible probability.

2.2 Mathematical Background

Before we describe in detail the concept of pseudo-free Groups, we present some mathematic background which is closely related to notion of pseudo-free groups.

2.2.1 Definition 1 Group[31] - A mathematical group G is defined as a pair of *(S,o)* where S is the set of elements and 'o' is an operator on S that satisfies the following properties.

- ***Closure***- This property states that if there exists two elements *i,j* that belongs to set S and defined by some binary relation o then the resultant $(i\ o\ j) \in S$.
- ***Identity*** - This property states that if there exists an identity element $e \in$ S such that for all i in S, $i\ o\ e = e\ o\ i = i$.
- ***Associativity*** - It states that for some elements say *a,b,c* that belongs to set S and defined by some binary operator o then
$i\ o(j\ o\ k) = (i\ o\ j)\ o\ k$.
- ***Inverses*** - It states that for some element $i \in S$,there exists some other element $j \in S$ such that the following condition holds:
$(i\ o\ j) = (i\ o\ j)$.

The notation xy is same as $x \ o \ y$ and $a's$ inverse is denoted by a^{-1}. The notation x^n means that the word $(x, x, x, ..., x)$ is of length n and x^{-n} is the word $(x^{-1}, x^{-1}, ..., x^{-1})$ of length n.

2.2.2 Computational Groups

Informally, a mathematical group G has some representation $[G]$ and such representations $[G]$ are known as computational groups[31]. In cryptography these groups are widely used and many other groups are based on the notion of such groups. For instance if we define set G as a group then we can consider G as a computational group $[G]$ with identity 1.
Formally such groups are defined as:

Definition 2[31]: *A group is said to be computational group if it represents an underlying mathematical group $G = (S, o)$ and provides efficient (polynomial time) algorithms for each of the operations given below:*

- **Composition**- It states that giving a representation of elements say $[i], [j]$ that belongs to computational group $[G]$, compute representation of $[i]o[j]$
- **Identity**- Find representation of identity element e that belongs to $[G]$
- **Inverses**-Given some element $[i]$ that belongs to $[G]$,find $[i^{-1}]$.
- **Equality Testing**- It states that on giving representation of two elements [i][j] that belongs to computational group$[G]$,determine if $i = j$
- **Sampling**- (only if G is finite) returns representation $[i]$ that is chosen uniformly at random from $[G]$ or in a manner that is similar from uniformly at random to a probabilistic polynomial time adversary. Such a procedure is denoted as $[i] \in_R [G]$.

2.2.3 Black Box Groups

Babai and Szemeredi introduced a notion- Black Box. This notion may be used by the parties in cryptographic protocol to access the group. According to black box assumption [31], each element of the computational group G is a bit string of some common length N, and Black Box subroutines are available for the group operations.

Only with the help of given implementations group operations can be performed. This is Black Box assumption, in which the representation of group elements is opaque: operations on them other than through the black box routines are forbidden.

Naturally, one may be curious to enquire the possibility for having black box algorithm for various group-theoretic problems. As the selections of the operations to be performed during algorithm design is restricted, Black Box assumption results in self imposed restrictions or conventions which are in turn favorable for algorithm design. Various techniques are not required for finding an algorithm if we use black box assumption as it itself gives satisfying result.

For example A probabilistic black box algorithm given by Tonelli nad shanks for finding the square roots of Z_P^* . This algorithm is used to find representation of an element $[x]$ having value x on giving $[a]$ and P such that:

$$x^2 = a \bmod P$$

The black box assumption becomes restrictive if there does not exist any efficient algorithm to compute this. For example shoup has proved lower bounds for discrete algorithm and other problems in discrete model.

However our priority for studying is cryptographic security not algorithmic efficiency. An adversary may check the code that implements group operations or test the bits of any representation as it is likely for typical adversary to violate any black box assumption willfully.

As an example like Z_N^* where the adversary is allowed to see N and a code for composition. Therefore the attacker can see the code or bit-level representation of elements by using techniques like index method calculus which is not there in black box assumption.

Thus, we do not consider black box assumptions. We assume by default that adversary can access any obtainable information and also may use techniques which are dependent on implementation or representation details. Non-black-box admission of the adversary to the group implementation is thus considered universally. A representation of a Computational groups allows one to decide whether a group is pseudo-free or not. Therefore one can think like that if an equation has a solution in computational group , then only there exists a solution of the corresponding equation in a mathematical group.

2.3 Division intractable functions

A function H is said to be division intractable[9] if it is not possible for an adversary to compute $(\{p_1, ..., p_n\}, q)$ such that $q \neq p_i$ for all $(i = 1 \text{ to } n)$ and $H(q)$ divides $H(p_i)$. The notion of division intractable functions was introduced by Gennaro et al. Any function that maps input to prime numbers clearly satisfies the property of division intractable functions.

Definition 3 [9] - Division intractable functions

Let $\mathcal{H}$ denotes family of hashes having poly(h)-bit input and h-bit output. A function H is said to be division intractable if it is not possible for an adversary to compute

1) H is randomly chosen from $\mathcal{H}$
2) Adversary produces $(\{p_1, ..., p_n\}, q)$ such that $q \neq p_i \forall$ (i = 1 to n) and $H(q)$ divides $H(p_i)$.

2.4 Shamir's Trick

Assume $x^a = y^b modN$ for all a,$b > 0$.Let $z = gcd(a$,$b)$, Then there exists an algorithm that computes f such that $f^a = y^z modN$.

Proof:- Using Extended euclidean algorithm we can find u,$v \in Z$ such that au+$bv = z$.
$\Rightarrow bv = z - au$.
Thus $x^{av} = y^{bv} = y^{z-au}$
$\Rightarrow y^z = x^{av}y^{au} = (x^v y^u)^a = (f)^a$

3 Free Groups

Free groups are infinite groups that are constructed from a given set of generators such that there do not exist any non trivial relationship between them.
The formal definition of Free group[31] is as follows:

Definition 4: *Let* $A = \{a_1, ..., a_m\}$ *represents a non-empty set of distinct symbols such that* $\{a_1, ..., a_m\}$ *are known as generators of the free group. Let the inverse of each symbol say* a_i *is given by* a_i^{-1} *such that then* A^{-1} *represents the non-empty set* $\{a_1^{-1}, ..., a_m^{-1}\}$*. Let* $\hat{A}$ *denotes* $A \bigcup A^{-1}$ *where as* $\hat{A}$ *is the set of symbols for the free group .*
Let the free group is denoted by $F(A)$, it can also be written as $F(a_1, ..., a_m)$ if $A = \{a_1, ..., a_m\}$. Elements in free groups can be written as words.

For example the word below:

$$a_1^{-1}a_1a_2^{-1}a_3a_3^{-1}a_3a_4$$

is representing elements of $F(a_1, a_2, a_3, a_4)$

A word in a free group can be reduced by eliminating the adjacent inverse symbols. Therefore the word in the given above example can be reduced and written as $a_2^{-1}a_4$. A word is said to be in reduced or canonical form if it cannot be reduced further. Thus we can say that the words in canonical form are the elements of the free group.
The binary operation o in a free group is a concatenation followed by simplification.

For example

$$a_1a_2 \ o \ a_3^{-1}a_1^{-1} = a_1a_2a_3^{-1}a_1^{-1} = a_2a_3^{-1}.$$

An empty word is the identity of a free group. Two words in a free group represent the same element if their canonical forms after simplification are exactly same. The reverse of the word is known as the inverse of the word.

3.1 Free Abelian Groups

A group is said to be free abelian group [31,19]if it is defined in a same manner as a free group. These groups are denoted as $FA\{a_1, ..., a_m\}$ such that $\{a_1, ..., a_m\}$ are known as generators.

The reduced form of a word in free abelian group $FA\{a_1, ..., a_m\}$ to a word of the form :

$$a_1^{s1}a_2^{s2}a_3^{s3}....a_m^{s_n}$$

is defined by commutativity whereas $s_1, s_2, s_3, ..., s_n$ are integers.

A free abelian group $FA\{a_1, ..., a_m\}$ is isomorphic to the m-fold direct sum Z$\bigoplus$ Z$\bigoplus$...$\bigoplus$Z. Thus we can represent an element $a_1^{s1}a_2^{s2}a_3^{s3}....a_m^{s_n}$ of $FA\{a_1, ..., a_m\}$ with vector $(s_1, s_2, s_3...., s_n)$ and implement o with the vector addition.

3.2 Equations in Free Groups

The equations in free groups[31] are as follows:
Let 'F' be a free group having a set of generators $a_1, a_2, a_3, ..., a_m$. Let $x_1, x_2, x_3, ..., x_n$ be variables in F. Thus an equation in free group F can be written as:

$$w_1 = w_2$$

where w_1 and w_2 represents the words formed from variables of F i.e. $x_1, x_2, x_3, ..., x_n$ and set of generators $a_1, a_2, a_3, ..., a_n$. We can also write these equations in reduced (canonical) form for some w.

For example an equation in $F\ (a_1, a_2)$

$$a_1^{-1}x_1 = x_2a_2$$

In free groups the equations that have solutions are known as satisfiable equations and the equations that do not have any solution are known as unsatisfiable equations. Makanin in 1982[24] showed that it is possible to find whether an equation is satisfiable or unsatisfiable in a free group. More recently Gutierrez in 2000[16] showed that the above problem is possible in PSPACE.

When the free group is abelian group $FA(a_1, a_2, a_3, ..., a_n)$, then one can find whether the equation has solution or not. The equation is always rewritten as:

$$x_1^{e_1}x_2^{e_2}...x_n^{e_n} = a_1^{s_1}a_2^{s_2}...a_m^{s_m}$$

for integers $e_1, e_2, , e_n$ and $s_1, s_2, ..., s_m$. These equations are satisfiable only if $\forall i, 1 \leq j \leq m$ we have

$$\gcd(e_1, e_2, ..., e_n)|s_j$$

An equation which has solution in $F(A)$ has also solution in $FA(A)$ but the converse is not always true. This is beneficial because it gives an easy way to show that an equation has no solution in a $F(A)$ and merely prove that it has no solution in $FA(A)$.

3.3 The correspondence

Pseudo-free groups considered to be a strong notion from cryptographic perspective and many other cryptographic assumptions holds on to the notion of pseudo-freeness. In each and every case one can find its formulation interesting.
For example[31], for the below quadratic equation

$$x^2 = a$$

there does not exist any solution in a free group because the canonical form of x always has even length where x is variable and a is the generator.

Thus the above equation has no solution in a free group, what does it mean for the same equation in group G?
In computational groups there may exist solutions of non-trivial equations. For example there exist a solution of the above mentioned equation in Z_N^* such that.

$$x^2 = a$$

has a trivial solution if one is given $a = 4$ then we can find a clear solution as $x = 2$. The above problem can be solved if each of the generators say a_i must correspond independently to a randomly generated element of G. As in [31], an adversary can distinguish a group G from a free group if

(1) An adversary outputs an equation E with variables $x_1, ..., x_n$ and constants $a_1, ..., a_m$ that has no solution in a fre e group.

(2)An adversary come up with randomly chosen values $a_1, ..., a_m$ and use them as constants for the equation.

(3)An adversary suceessfully produce variables $x_1, ..., x_n$ that satisfiies the equation.

As per definition of pseudo-free groups, the adverary can selects equation by his own as long as the equation has no solution in a free group. This generalize the situation for the strong RSA assumption where the adversary can select the exponent.

3.4 Pseudo-free Gropus

More specifically , a computational group is said to be pseudo-free group if it is not possible for an adversary who is given only the group description to produce an equation E and solution S to that equation. For example in pseudo-free groups, if an adversary is given randomly a then it is not possible for him to efficiently compute solution of an equation $x^e = a$ where $e \neq 1$, or for $x_1^3 x_2^4 = a^5$, but can solve an equation $x_1^2 x^2 = a^5$ because this equation is trivial and has solution over the free group if we put $x_1 = a^2$ and $x_2 = a$.

Definition 5[31,26].:- *A family of computational groups $G = G_N$ for $N \geq 0$ is said to pseudo-free if all operations in G_N is performed in polynomial time in k. and for every probabilistic polynomial time adversary $\mathcal{A}$ following holds. Let $N \in N_k$ be a randomly chosen group index and define $\alpha : A \to G_N$ by choosing $\alpha(a)$ at random in G_N for each $a \in A$. Then the probability on giving (N,α) as input, the adversary outputs an equation 'E' and solution 'S' is almost negligible in k.*

This definition belongs to the family of computational groups but we can apply this definition to mathematical groups by understanding that the implementation of such groups in some standard way as computational groups.

3.4.1 Types of pseudo-free groups

There are two types od pseudo-free groups which are as follows
(1) Static pseudo-free groups
(2) Adaptive pseudo-free groups
Static pseudo-free groups:-
A group is said to be static pseudo-free if an adversary who is allowed to see description of the group can not solve new nontrivial equation and solution.
Adaptive pseudo-free groups:-
In adaptive pseudo-free groups, along with group description an adversary is allowed to see some non-trivial equations and solutions and finally his goal is to come up with new non-trivial equation and solution.

3.5 Cryptographic assumptions holds on pseudo-free groups

There are six cryptographic assumptions[31] considered by Lipschutz and Miller that are as follows:

- The order problem
- The power problem
- The root problem

- The proper power problem
- The generalized power problem
- The intersection problem

Lipschutz and Miller has shown that the above mentioned six fundamental problems are independent i.e. for each pair of problems there exists a group such that one problem has solution in a group where as other problem has no solution.

Rivest[31] has shown that the above six fundamental problems are intractable in pseudo-free groups such that no probabilistic polynomial time adversary can solve these problems with non-negligible probability. On the other side, Rivest also left a very interesting question whether or not the computational Diffie-Hellman problem (CDH) holds true by pseudo-free groups.

Order problem

The order problem states the following: Given an input $x \in G$ to find an integer n which satisfies

$$x^e = 1$$

where e is a positive integer and 1 is the identity of group. Thus it means that solving $x^e = 1$ for positive integer n. The order of the element x is the least positive value of x in G.

Theorem 1 ([31]): *In a pseudo-free group G, it is hard for an adversary to state the order of an element p that is chosen randomly.*

Power problem

The power problem states the following:
Given elements x and y such that $x, y \in G$ to find an integer $n > 1$

$$x^n = y$$

such that the value of n is the discrete logarithm of y to the base x in G. The power problem is often considered to be hard for some groups G. For example Diffie and Hellmann determined that the power problem was hard in Z_e^* where e is a large prime.

Theorem 2([31]) *In a pseudo-free groups, it is hard for an adversary given some random values x and y to compute the discrete logarithm problem.*

Root problem(RSA assumption)

The RSA problem states the following:
Given input $y \in G$ and an integer e greater than 1 to find x that satisfies

$$x^e = y$$

Theorem 3:([31])*In a pseudo-free groups, the RSA assumption holds.*

Proper power problem(strong RSA assumption)

The RSA problem states the following:
Given input $y \in G$ to find an integer e greater than 1 and x that satisfies

$$x^e = y$$

Theorem 4:([31]) *In pseudo-free groups, the strong RSA assumption holds.*

In this case the adversary by himself selects an exponent $e > 1$. The adversary cannot solve the above equation even if he comes up with an integer e >1 on his own.

Generalized Power problem

The generalized problem is defined as:
Given inputs $p, q \in G$ to get integers x, y that satisfies

$$p^x = q^y$$

Theorem 5:([31])*In pseudo-free groups, it is hard for an adversary to compute the generalized power problem.*

Intersection Power problem

The intersection problem is defined as:
Given inputs $p, q \in G$ to get integers x, y that satisfies

$$p^x = q^y \neq 1$$

Theorem 6:([31])*In pseudo-free groups, it is not possible for an adversary to compute the intersection problem for cyclic subgroups.*

3.6 Diffie Hellman assumption

Fascinatingly, the(computational) Diffie-Hellman problem is unsuitable for our formalism. Diffie-Hellman assumption[31] is implied by pseudo-freeness is a very appealing open problem. More specifically it is left as an open problem by Rivest whether Diffie Hellman assumption holds the property of pseudo-freeness or not.

The computational Diffie-Hellman problem(CDH)[31] considers the following

Given some value g and two other values x and y such that

$$x = g^i$$

$$y = g^j$$

for integers i and j that are chosen randomly to find

$$a = g^{ij}$$

With the provided x and y, an adversary will have slight chance for computing a, this is CDH assumption. The most obvious way to prove that the CDH assumption is implied by pseudo-freeness is via above equations, where$i$ and j are integer-valued variables, and a is a group valued variable. However, this argument does not succeed as an adversary who violates CDH to deduce a need not be able to find i and j (this is DLP). It appears that any equation in variable a alone, for checking the correctness of computed a is unavailable. In other words the decisional Diffie-Hellman problem is not satisfiable by checking a set of equations that involves a single element x.

4 Static pseudo-free groups

The notion of pseudo-free groups to static adversary has been extended by D.Micciancio[26]. Informally static pseudo-free groups are the computational groups where the adversary is given only randomly chosen generators of the equation and his goal is to come up with both equation and solution. More specially, in these groups the adversary is static i.e. allowed to see only description of the group without any other information. If the order of the group is known then it is very easy to compute solution for nontrivial equation Thus in static pseudo-free group, order of the group is hidden from adversary.

The definition given by Micciancio was similar to that one defined by Rivest in [31]. Micciancio proved that RSA groups holds much stronger version of pseudo-freeness. He has shown that it is infeasible for an adversary to compute unsolvable equations along with their solutions in a given computational group. His proof that RSA is pseudo-free in [26] is based on following properties with the assumption that starting from arbitrary equations yields simpler and simpler equations.

- In a free group unsolvable equations are mapped to unsolvable equations.
- Over the computational group, one can map solution of original equations to solution of resulting equations

4.1 Formal Definition

Definition 6 [26]- *A family of computational groups $G = G_N$ for $N \geq 0$ is said to be static pseudo-free if all operations in G_N is performed in polynomial time in k. and for every probabilistic polynomial time adversary $\mathcal{A}$ following holds. Let $N \in N_k$ be a randomly chosen group index and define $\alpha : A \rightarrow G_N$ by choosing $\alpha(a)$ at random in G_N for each $a \in A$. Then the probability on giving (N,α) as input, the adversary outputs an equation 'E' and solution 'S' is almost negligible in k.*

The definition of pseudo-free groups given by Rivest was for the single equation which has no solution in a free group but has solution in computational group. Micciancio made extension to the definition of pseudo-free groups that they consist of single equation to a system of equations by transforming system of equations to a single equation which is equivalent to the system. Any system of equation in this transformation can map to single equation whose solution set is a superset of the solutions to the system.

Definition 7 [26]- *A family of computational groups $G = G_N$ for $N \geq 0$ is said to be static pseudo-free if the the probability on giving (N,α)= $({E^i}_{\{i \in I\}}, \psi)$ as input to an adversary, the adversary outputs an system of equations ${E^i}_{\{i \in I\}}$ with their solutions $\psi : X \rightarrow G_N$ is almost negligible in k.*

4.2 The RSA is static pseudo-free

Lemma 1[26]:- *For a family of computational groups $\mathcal{G}$ there exists an PPT algorithm that on input an equation Λ (which is a pair of $\{X,A\}$ such that 'X' is set of variables an 'A' is set of constants), a group G and an assignment δ:X $\rightarrow$G, outputs a single variable equation $\Lambda'$and solution $\delta' \in$ G that satisfies the following properties:*

(1) If an equation Λ does not have a solution in a free group, then Λ' also has no solution in a free group.

(2) For any assignment a:A $\rightarrow$ G, if δ is a solution to Λ_a, then δ' is a solution to Λ'

Proof:- Let

$$\Lambda = \{x_1^{e_1}x_2^{e_2}....x_n^{e_n} = a_1^{s_1}a_2^{s_2}....a_m^{s_m}\}$$

be an input equation and δ:X $\rightarrow$G be an variable assignment.

Thus by using extended euclidean algorithm one can find$e = gcd(e_x : x \in X)$ and integers $e'_x(x \in X)$ such that

$$\textstyle\sum_x e_x e'_x = e$$

Therefore the algorithm on input the above equation outputs an single variable equation which is of the form:

$$\Lambda' = \{x^e = a_1^{s_1}a_2^{s_2}....a_m^{s_m}\}$$

along with its solution $\delta' = \prod_{x \in X} \delta(x)^{e_x/e}$. Thus we have to prove that the output equation satisfy the above properties

In order to prove first property,for sake of contradiction we assume that there exists a solution of the the output equation Λ' in a free group. Let δ' be a such solution. then we have

$$\delta'^e = a_1^{s_1}a_2^{s_2}....a_m^{s_m}$$

Thus for any $x \in X$ we have $\delta(x)^{e_x/e} = (\delta')$

$$\Rightarrow \delta(x)^{1/e'_x} = (\delta')$$

$$\Rightarrow \delta(x) = (\delta')^{e'_x}$$

We can clearly see that δ is a solution of equation Λ because

$$\textstyle\prod_{x \in X} \delta(x)^{ex} = \delta'^{\sum_x e_x e'_x}$$

$$\Rightarrow \delta'^{e} = a_1^{s_1} a_2^{s_2} .... a_m^{s_m}$$

This proves the first property which means that the output equation δ' also has a solution in a free group.

To prove second property, first of all set an assignment $a : A \rightarrow G$ and let $\delta : X \rightarrow G$ be asolution to Λ_a which means that

$$\textstyle\prod_{x \in X} \delta(x)^{e_x} = a_1^{s_1} a_2^{s_2} ... a_m^{s_m}$$

Therefore

$$\delta'^{e} = (\textstyle\prod_{x \in X} \delta(x)^{e_x/e})^{e} = \textstyle\prod_{x \in X} \delta(x)^{e_x} = a_1^{s_1} a_2^{s_2} .... a_m^{s_m}$$

Thus we can see that δ is a solution to Λ_a over G.

5 Adaptive pseudo-free groups

In adaptive pseudo-free groups[9] an adversary is permitted to look at some nontrivial equations and solutions along with the group description and finally his goal is to come up with new non-trivial equation and solution. As discussed above that in case of static pseudo-free groups the adversary is given only the randomly chosen generators used in equation and he has to come up with new nontrivial equation and solution. Therefore in our adaptive setting the adversary knows additional information which may help him in solving other equations which do not have a solution in a free group. In case of adaptive pseudo-free groups the game between challenger and adversary works as follows[9]:

Set up:- In this case the challanger randomly selects an instance of compuational group,sets an assignment and then gives assignments and group description to adversary.

Equation queries:- In this phase the adversary can see some equations and solutions

Challenge:- In this stage the adversary is given a chllenge that he has to produce a new non-trivial equation with its solution.

The definition of pseudo-free groups in adaptive setting[9] consists of two design decisions. The first issue with the design is to set equations for which an adversary can see solutions and the second issue is to define what do we actually mean by non-trivial multivariate equations in adaptive setting of pseudo-free groups.
As discussed above the first issue is to fix the type of equations but the important thing which is to be considered here is how the equations can be produced for which adversary can see solution because if we give too much freedom to adversary then it can violate the security. Even keeping more restrictions here won't allow us to define adaptive setting for an adversary. Therefore first issue with the design has been resolved by considering some parametric distribution over the set of equations.

5.1 Equation setup in adaptive pseudo-free groups

In second stage of adaptive pseudo-free groups,the adversary is allowed to see some nontrivial equations and solutions. Therefore the equations are controlled by parametric distribution[9]. More specifically, equations are selected according to some parametric distribution where the distribution depends on the parameter of an adversary. The parameter supplied by adversary may lead to different variety of adversaries ranging from weak one to a strong one.

In case of weak adversary, we consider weak definition of adaptive pseudo-free groups where the adversary has no control over the equations. This means that the challenger chooses an equation , sets an assignment, computes the solution of an equation and finally gives equation and its solution to an adversary. Hence in this case the adversary is not that much adaptive, so he can get all equations together with their solutions at once.

In strongest form of adversary, the case is not the same as we saw in weakest form. In this case the adversary is really adaptive and can select equations by his own choice. The adversary can choose equations in an adaptive manner such that the adversary first choose an equation, get back its solutions, then choose another equation, get back solution and so on. This definition of pseudo-freeness is known as strong adaptive pseudo-freeness.

In static pseudo-free groups where the adversary is given only randomly chosen generators of the equation and his goal is to come up with solution to that equation. In contrast, the notion of pseudo-freeness in adaptive setting[9] has been extended to more general level. In case of adaptive adversary,we have done a new kind of setting where the adversary has limited control over the set of equations. This limitation is controlled by parametric distribution φ. The adversary supplied a parameter M of some length which is required for sampling from distribution. The distribution then provides $m+1$ integers which is written as (e, s) where as e and s are integers. Once the parameter M is fixed, then $\varphi(M)$ is the distribution from which e and s are drawn.

5.2 Non-trivial Multivariate Equations

Rivest in [31] defined non trivial equations in a way that they do not have any solution in a free group. We define non-trivial equations in some other way because in our case the adversary is allowed to see some additional information also which may be useful for him to solve other non-trivial equations. Therefore we define non-triviality by analysing equations over quotients of free groups.
we describe a framework that defines the non-triviality for multivariate equations.

Let [9] $FA(a_1...a_m)$ denotes a free abelian group such that $\{a_1...a_m\}$ are the generators of the group and there exists some binary relation $\Lambda \subseteq FA \times FA$ on free abelian group that makes equalities in words. As mentioned earlier, we would now hence aim to characterize the set of all equalities that can be derived from Λ . Recollect the fact that ultimately these equalities are construed over computational groups, thus there are two ways for an adversary to formulate new equalities Λ. The first way is by using group operations and their properties over computational groups.
For example let $\Lambda = a_1a_2 = a_1^2a_4$ then one can also find $a_1a_2^2 = a_1^2a_4a_2 = a_1^3a_4^2$.Thus we have computed first equality just by multiplying a_2 to both sides in the equation and second we have derived from the property of computational group. For instance if one can derive the equality in the form $w_1^b = w_2b$ in computational group, then can also find equality of the form $w_1 = w_2$ provided $'b'$ should be prime. Moreover,we have to

take into account the above possibility for any q as we search for an abstraction which is not dependent on the order of the group.

Definition 8[9]- *Let $FA\{a_1.....a_m\}$ be a free abelian group defined by some binary relation $\Lambda \subseteq FA \times FA$ on $FA\{a_1....a_m\}$. Let $\equiv_\Lambda$ be the smallest congruence on FA.*

- $\Lambda \subseteq \equiv_\Lambda$
$\forall p \in N, \forall w_1, w_2 \in FA, w_1^b \equiv_\Lambda w_2^b \Rightarrow w_1 \equiv_\Lambda W_2$.

then $w_1 and w_2$ are trivially equal w.r.t $\Lambda is w_1 \equiv_\lambda w_2$.
Therefore description for $\equiv_\Lambda$ is as follows.

Let $\Lambda = \{(w_1, 1, w_2, 1), (w_1, 2, w_2, 2), ..., (w_1, t, w_2, t)\}$. Consider a binary relation R_Λ on free abelian group defined by:$(w_1, w_2) \in R_\Lambda$ if $\exists l_1, l_2...l_t \in Q$ such that

$$w_1 = w_2.\prod_{i=1}^{t}(w_{1,i}^{-1}.w_{2,i})^{l_i}$$

Thus we can say that exponentiation of the word $w = a_1^{s_1}.....a_m^{s_m}$ with rational number $l = a/b$ is defined if and only if b divides $gcd_{1 \leq i \leq n} a.s_i$.

5.3 Applications of adaptive pseudo-free groups

Adaptive pseudo-free groups[9] has an application to signatures and network coding signatures. These applications has been defined for univariate equations.

5.3.1 Signatures from Adaptive pseudo-free Groups

Construction of signature scheme

A signature scheme for family of adaptive pseudo-free groups with univariate equations[9] works as follows:
Let φ^* denotes a parametric distribution that belongs to a specific class of parametric distribution φ_n and G be a family of adaptive pseudo-free groups with respect to φ^*.
Thus the signature scheme works as :

(1) Key Generation $(1^k, N)$

Let $A = \{a_1, a_2....a_m\}$ and $X = \{x\}$ be a set of constants and variables. In this phase first of all the key generation algorithm randomly choose an instance of computational group G_N from a family of computational groups $\mathcal{G}$ then sets an assignment $\alpha : A \to G_N$.
Finally it sets verificaion key(public key) $vk = \{X, A, \alpha, G_N, \varphi^*\}$ and signing key(secret key) as $sk = order\{G_N\}$.

(2) Signing (sk,M)

In this phase, φ^* gives (e, s, r) as output and the signing algorithm use secret key which is equal to order of G_N to compute solution of the univariate equation which is in the form of

$$x^e = a_1^{s_1}.....a_m^{s_m}$$

If ψ be a solution of the equation then the signing algorithm finally gives signature as $\sigma = (e, s, r, \psi)$ for M.

(3) verification(vk,M,σ)

In this phase the verification algorithm performs the verification of signature on message M by checking two conditions.

1)It verifies whether $ver_{\varphi^*}(e, s, r, M) = 1$.
2)It checks if ψ is a solution of the equation $x^e = a_1^{s_1}.....a_m^{s_m}$.

If both the conditions are satisfied than it gives 1 as output ,otherwise 0.

5.3.2 Linear Network Coding

Network coding is technique of transmitting information in which the nodes of the network combines several packets they receive and then transmits them in order to achieve maximum information flow in the network. The transmitted information[9] is expressed in a vector form as $(v_1, ..., v_m)$. Before transmitting any information, a sequence of m-augmented vectors$(p_1, ..., p_m)$ are generated by the source code by prepending u_i to v_i. Each u_i has 1 in i-th position and 0 in other positions. The source in the network sends the augmented vectors as packets in the network. The nodes in the network after receiving $p_1, p_2, ..., p_m$ process these packets by performing linear combination and then sends them as a resultant vector to its outgoing edges.
A node must receive m valid vectors pi of the form $p1, p2, ..., pm$ to get original information and for which $u_i s$ are linearly dependent. Thus the original message M is retrieved as

$$M = U^{-1}V$$

Where U and V are the matrix with rows $u_1, u_2, ..., u_m$ and $v_1, v_2, ..., v_m$.

The malicious nodes in the network makes the retrieval of the original information impossible by injecting invalid vectors. The above mentioned idea is susceptible to these kind of attacks. A solution to overcome this problem is to use network coding signatures that checks whether a given vector is valid or not just by checking condition whether it has been generated by linear combination of valid vectors.

5.3.3 Network Coding Signatures

Definition 9[9] : *A network coding signature is defined by a triple of algorithms (NetKG, Sign, Ver) such that*:

(1) NetKG**$(1^k$**,N)

In this phase the algorithm takes input as k and N and produces output vk and sk.

Where k is the security parameter
N is the parameter that defines the size of signed vectors
vk is the public verification key
sk is the secret signing key.

(2) sign(sk, V)

The signing algorithm in this phase produces signature σ for message M on input sk, fid and m dimensional subspace $V \subset FN$.
Where fid is a random file identifier.

(3) ver(vk, V,**σ**)

The verification algorithm on input $(vk, fid, V \subset FN,)$ gives output as 1 (accept) or 0 (reject). This algorithm checks one condition that if σ is valid signature produced by signing algorithm for all honestly generated key pairs (vk, sk), all fid (random file identifier) and for all $V \subset FN$. If the condition is satisfied then the algorithm gives output as 1 for all $v \in V$ otherwise 0.

5.3.4 Homomorphic Network Coding Signatures

Definition 10[9]:- : *Homomorphic network coding signatures are special type of network coding signatures that are defined by a 4-tuple of algorithms (NetKG, Sign, Ver, Combine)such that:*

(1) NetKG(1k, N)

In this phase the algorithm takes input as k and N and produces output vk and sk.
Where k is the security parameter
N is the parameter that defines the size of signed vectors
vk is the public verification key
sk is the secret signing key.

(2) Sign (sk, V, fid)

The signing algorithm in this phase produces signature σ for message M on input sk, fid and m dimensional subspace $V \subset FN$.
Where fid is a random file identifier.

(3) Combine (vk, fid,$(w_i, \sigma_i)_{i=1}^{l}$)

On input vk,fid (random file identifier), set of tuples (w_i, σ_i), this algorithm produce a new signature σ as output such that if each σ_i is a valid signature on v_i then σ is a valid signature for v which is obtained from linear combination $\sum_{(i=1)}^{l} w_i v_i$.

(4) Ver (vk, V, fid) -

The verification algorithm on input $(vk, fid, V \subset FN, \sigma)$, gives output as 1 means accept or 0 means reject. This algorithm checks one condition that if $sigma$ is valid signature produced by signing algorithm for all honestly generated key pairs (vk, sk), all $fid's$ (random file identifier) and for all $V \subset$ FN . If the condition is satisfied the algorithm gives ouput as 1 for all $v \in V$ otherwise 0.

5.3.5 Network Coding Signatures from Adaptive pseudo-free groups

As we have discussed above in[9] network coding signature scheme, a file to be transmitted is expressed as an ordered sequence of set of vectors $(v_1, v_2, .v_m)$ of n components. These vectors will be prepended with a vector u_i of m components. Let w_i be the set of resulting vectors.
Let $R = \{0, ..., r-1\}$ denotes the set of coefficients from where they are randomly sampled. Let U denotes an upper bound on the path length from source to destination.Thus from the above positions $B = mr^U$. This denotes the largest value of u-coordinates in vectors. Moreover denoting an upper bound with M on the magnitude of the coordinates of the initial vectors $\{v1, v2, ..., vm\}$, we set $B^* = MB$.
Let φ_n denotes the parametric distribution that takes input as (fid, V, B^*).Let l_s denotes a security parameter and l represents an integer such that $2^l > B$, compute $H(fid)$ where H is an division intractable function. For each $v_i = (v_1^i, v_2^i, ..., v_n^i) \in V$. First of all it uniformly samples $al + l_s$ bit random integers s_i and then gives output as (s_i, u^i, v^i). Thus φ_n gives output as $(e, \{s_i, u^{(i)}, v^{(i)}\}_{i=1}^{m}$

Let φ^* denotes a parametric distribution belongs to a specific class of parametric distribution φ_n and G be a family of adaptive pseudo-free groups with respect to φ^*. Thus the signature scheme works as :

(1) NetKG*(1^k*,N)

Let $A = p_1, ..., p_n, q_1, ..., q_2$ and $X = x$ be a set of constants and variables. In this phase first of all the key generation algorithm randomly choose an instance of computational group G_N from a family of computational groups $\mathcal{G}$,then sets an assignment $\alpha : A \rightarrow G_N$.
Finally it sets verificaion key(public key) $vk = \{X, A, \alpha, G_N, \varphi^*\}$ and signing key(secret key) as $sk = order\{G_N\}$.

(2) sign(sk, V)

In this phase, first of all the signing algorithm choose a random indentifier fid for V and then runs φ_n to obtain $(e, \{s_i, u^{(i)}, v^{(i)}\}_{i=1}^m$ Finally it uses secret key which is equal to order of G_N to compute solution of the univariate equation which is in the form of

$$x^e = p^{s_i} p_1^{v_1^i} ... p_n^{v_n^i} q_1^{u_1^i} ... q_2^{u_m^i}$$

If ψ be an solution of the equation then the signing algorithm finally gives signature as $\sigma = (e, \{s_i, u^{(i)}, v^{(i)}\}_{i=1}^m, \psi)$ for M.

(3) ver(vk, V,*σ*):

In this phase the verification algorithm performs the verification of signature on message M by checking two conditions.
1)It verifies whether $ver_{\varphi_n}(e, \{s_i, u^{(i)}, v^{(i)}\}_{i=1}^m, \psi) = 1$.
2)It checks if ψ is a solution of the equation $x^e = p^{s_i} p_1^{v_1^i} ... p_n^{v_n^i} q_1^{u_1^i} ... q_2^{u_m^i}$.

If both the conditions are satisfied than it gives 1 as ouput ,otherwise 0.

(4) combine(vk,fid,*$\sigma_1, ..., \sigma_l$*):

This algorithm works as follows:
First of all it removes w_i that have u co-ordinates negative or larger than $B/(mq)$ or have v coordinates negative or larger than $B^*/(mq)$. Thus the remaining vectors are $w_1, w_2, .., w_n$.

Finally this algorithm randomly choose $a_1, a_2, ..., a_n \in Q$ then set $w = \sum_{i=1}^n a_i w_i$ and produce signature as output $\sigma = (e, s, w, fid, \psi)$ on w by computing

$\psi = \prod_{i=1}^n \psi_i^{a_i}$
$s = \sum_{i=1}^n a_i^{s_i}$

5.4 Trivial Multivariate Equations

Let

$$\Lambda = \{x_1^{e_1^k} x_2^{e_2^k} ... x_n^{e_n^k} = a_1^{s_1^k} a_2^{s_2^k} ... a_m^{s_m^k}\}_{k=1}^{t}$$

be a set of multivariate equations over the free abelian group 'F' and let $\{\phi_{1,k}, \phi_{2,k}, ..., \phi_{n,k} | \kappa = 1.....\text{t}\}$ be a solutions for these equations. Assume the free abelian group 'F' is generated by $\{\phi_1, \phi_2, ..., \phi_n\ , \alpha_1, \alpha_2, ..., \alpha_n\}$.

Definition 10 : *Equation* $= \{x_1^{e_1^*} x_2^{e_2^*} .....x_n^{e_n^*} = a_1^{s_1^*} a_2^{s_2^*} ... a_m^{s_m^*}\}$

is trivial with respect to Λ *if there exists solution of the given equation over* $F/\!\equiv_\Lambda$.

Let

$$\{x_1^{e_1^*} x_2^{e_2^*} ... x_n^{e_n^*} = a_1^{s_1^*} a_2^{s_2^*} ... a_m^{s_m^*}\} \tag{1}$$

be an equation

Assume that

$$\phi_i^* = \prod_{y=1}^{m} a_y^{v_y^i} \prod_{y=1}^{n} \prod_{l=1}^{t} \phi_{y,l}^{k_{l,y}^i}$$

is a solution for the equation for some $v_y^i, k_{l,y}^i$ for $(1 \leq y \leq m, 1 \leq l \leq n, 1 \leq i \leq n)$ and rationales $l_1, l_2....l_t$ for some $\mathbf{Q}$ such that:

$$\{(\phi_1^{k_1^1}, ..., \phi_1^{k_n^t}, a_1^{v_1^1} ... a_1^{v_1^k})^{e_1^*} ....(\phi_n^{k_1^1}, ..., \phi_n^{k_n^t}, a_m^{v_m^1} ... a_m^{v_m^k})^{e_n^*}\} = a_1^{s_1^*} a_2^{s_2^*} .....a_m^{s_m^*} \prod_{i=1}^{t} (\prod_{y=1}^{n} \phi_{y,i}^{-e_y^i} \prod_{y=1}^{m} a_y^{s_y^i})^{l_i}$$

$$(\prod_{i=1}^{n} \phi_i^*)^{e_i^*} = a_1^{s_1^*} a_2^{s_2^*} .....a_m^{s_m^*} \prod_{i=1}^{t} (\prod_{y=1}^{n} \phi_{y,i}^{-e_i^y} \prod_{y=1}^{m} a_y^{s_y^i})^{l_i} \tag{2}$$

Thus by replacing the expressions in the above relation and matching exponents of different symbols we have an equation:

$$\{x_1^{e_1^*} x_2^{e_2^*} ....x_n^{e_n^*} = a_1^{s_1^*} a_2^{s_2^*} ....a_m^{s_m^*}\} \tag{3}$$

is trivial with respect Λ to if there exists $v_y^i, k_{l,y}^i$ for $(1 \leq y \leq m, 1 \leq l \leq n, 1 \leq i \leq n)$ and rationals $l_1, l_2....l_t$ for some $\mathbf{Q}$ such that

1) $\sum_{i=1}^{n} k_{u,y}^i e_i^* = e_y^u l_u$ for $(1 \leq u \leq t, 1 \leq y \leq n)$

2) $\sum_{i=1}^{m} v_y^i e_i^* = s_y^* - \sum_{u=1}^{t} s_y^u l_u$ for $(1 \leq y \leq m)$

The converse of the above statement also true if there exists if there exists $v_y^i, k_{l,y}^i$

for $(1 \leq y \leq m, 1 \leq l \leq n, 1 \leq i \leq n)$ and rationals $l_1, l_2....l_t$ for some $\mathbf{Q}$ such that equation (2) holds then

$$\phi_i^* = \prod_{y=1}^{m} a_y^{v_y^i} \prod_{y=1}^{n} \prod_{l=1}^{t} \phi_{y,l}^{k_{l,y}^i}$$

is a solution for the equation (1) over F/$\equiv_\Lambda$

Given the set of equations

$$\Lambda = \{x_1^{e_1^k} x_2^{e_2^k}....x_n^{e_n^k} = a_1^{s_1^k} a_2^{s_2^k}....a_m^{s_m^k}\}$$

we define the following quantities in a matrix form such that these quantities are dependent on Λ

$$\sum = \begin{bmatrix} s_1^1 & \cdots & s_1^t \\ \vdots & & \vdots \\ s_m^1 & \cdots & s_m^t \end{bmatrix}$$

$$E = \begin{bmatrix} 1/e_1^1 & \cdots & 1/e_1^t \\ \vdots & & \vdots \\ 1/e_n^1 & \cdots & 1/e_n^t \end{bmatrix}$$

We write the given condition in matrix form as

$$\begin{bmatrix} v_y^1 & \cdots & v_y^n \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix} = s_y^* - \begin{bmatrix} s_y^1 & \cdots & s_y^t \end{bmatrix} (l_u)$$

$$\Rightarrow \begin{bmatrix} v_y^1 & \cdots & v_y^n \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix} = s_y^* - \begin{bmatrix} s_y^1 & \cdots & s_y^t \end{bmatrix} \begin{bmatrix} k_{u,y}^1/e_y^u & \cdots & k_{u,y}^n/e_y^u \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix}$$

Proposition 1: *Equation*$\{x_1^{e_1^*} x_2^{e_2^*}....x_n^{e_n^*} = a_1^{s_1^*} a_2^{s_2^*}....a_m^{s_m^*}\}$
is trivial with respect to Λ *if and only if*

$$(1) \exists k \in Z^t, V \in Z^m : e_i^* (\sum EK + V) = s^*$$

$$(2) \forall m \in (1...t), \forall x, y \in (1...n), \exists l_m : \sum_{i=1}^{n} k_{m,x}^i (\frac{e_i^*}{e_x^m}) = \sum_{i=1}^{n} k_{m,y}^i (\frac{e_i^*}{e_y^m})$$

where $e_i^* = [e_1^*, e_2^*.........e_n^*]^T$
$s^* = [s_1^*, s_2^*.........s_m^*]^T$

Proof: The proof for above proposition can be done by setting

$$l_u = \sum_{i=1}^{n} k_{u,y}^{i} \frac{e_i^*}{e_y^u}$$

For example if we put ($u = 1$ and $y = 1$)in above condition then

$$l_1 = \sum_{i=1}^{n} k_{1,1}^{i} (\frac{e_i^*}{e_1^1})$$

Now let ($u = 1$ and $y = 2$) then

$$l_1 = \sum_{i=1}^{n} k_{1,2}^{i} (\frac{e_i^*}{e_2^1})$$

Thus we can say that

$$\forall m \in (1...t), \forall x, y \in (1...n), \exists l_m : \sum_{i=1}^{n} k_{m,x}^{i} (\frac{e_i^*}{e_x^m}) = \sum_{i=1}^{n} k_{m,y}^{i} (\frac{e_i^*}{e_y^m})$$

We write the above condition

$$e_i^* (\sum EK + V) = s^*$$

in matrix form as:

$$e_i^* (\begin{bmatrix} s_1^1 & \cdots & s_1^t \\ \vdots & & \vdots \\ s_m^1 & \cdots & s_m^t \end{bmatrix} \begin{bmatrix} k_{1,1}^1/e_1^1 & \cdots & k_{1,1}^n/e_1^t \\ \vdots & & \vdots \\ k_{t,n}^1/e_n^1 & \cdots & k_{t,n}^n/e_n^t \end{bmatrix} + \begin{bmatrix} v_1^1 & \cdots & v_1^n \\ \vdots & & \vdots \\ v_m^1 & \cdots & v_m^n \end{bmatrix}) = \begin{bmatrix} s_1^* \\ \vdots \\ s_m^* \end{bmatrix}$$

$$\begin{bmatrix} v_1^1 & \cdots & v_1^n \\ \vdots & & \vdots \\ v_m^1 & \cdots & v_m^n \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix} = \begin{bmatrix} s_1^* \\ \vdots \\ s_m^* \end{bmatrix} - \begin{bmatrix} s_1^1 & \cdots & v_1^t \\ \vdots & & \vdots \\ s_m^1 & \cdots & v_m^t \end{bmatrix} \begin{bmatrix} k_{1,1}^1/e_1^1 & \cdots & k_{1,1}^n/e_1^t \\ \vdots & & \vdots \\ k_{t,n}^1/e_n^1 & \cdots & k_{t,n}^n/e_n^t \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix}$$

Thus for $(1 \leq u \leq t, 1 \leq y \leq n) and (1 \leq y \leq m)$ we have,

$$\Rightarrow \begin{bmatrix} v_y^1 & \cdots & v_y^n \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix} = s_y^* - \begin{bmatrix} s_y^1 & \cdots & s_y^t \end{bmatrix} \begin{bmatrix} k_{u,y}^1/e_y^u & \cdots & k_{u,y}^n/e_y^u \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} v_y^1 & \cdots & v_y^n \end{bmatrix} \begin{bmatrix} e_1^* \\ \vdots \\ e_n^* \end{bmatrix} = s_y^* - \begin{bmatrix} s_y^1 & \cdots & s_y^t \end{bmatrix} (l_u)$$

$$\Rightarrow \sum_{i=1}^{m} v_y^i e_i^* = s_y^* - \sum_{u=1}^{t} s_y^u l_u$$

Adaptive Pseudo-free Groups

Informally, adaptive-pseudo-free groups[9] are the computational groups in which the adversary is permittef to look at several equations along with their solutions and his goal is to solve a new non-trivial equation where non-trivial equations means the equations that do not have solutions in a free group.
The formal definition of adaptive pseudo-free groups[9] given below is for a set 'A' of α generators, a computational group $\{G_N\}_N$ and a distribution ϕ over set of equations .

Setup
In this phase first of all the challenger randomly selects instance of computational group $\{G_N\}$ that belongs to a family of $\mathcal{G}$ where $\mathcal{G} = \{G_N\}_{N \in Nk}$ and then sets an assignment n: A→ $\{G_N\}$ for set A of generators and parametric distribution φ. At last the input to the assignment n: A→ $\{G_N\}$,description of the computational Group $\{G_N\}$ and parametric distribution φ is given to the Adversary $\mathcal{A}$.

Equation queries
In this phase the adversary is permitted to look at the equations along with their solutions. The queried equations are controlled by the adversary through parametric distribution φ. Thus the adversary $\mathcal{A}$ selects $\mathcal{M}_k$ for each query and gives it to challenger. Then the challenger runs $(e_i^k, s_j^k) \leftarrow \hat{\varphi}(\mathcal{M}_k)$, computes the solution ψ_i for equation λ_i and gives back (ψ_i,(e_i^k, s_j^k)) to adversary $\mathcal{A}$.

Challenge
After the adversary is acquaimted with equations and their solutions then he tries to produce an equation λ^* with a solution ψ^*.The adversary succeeds in the game if he successfully produces a non trivial equation.
Definition 11 [9] Adaptive pseudo-free Groups:- *A family of computational Group $\mathcal{G}$ is said to be adaptive Pesudo-free with respect to some distribution φ, if for any set 'A' of polynomial size, any probabilistic time adversary $\mathcal{A}$ succeeds in the game above with at most negligible probability.*

5.5 The RSA group is Adaptive Pseudo-free

Theorem 7 :- *If the strong RSA assumption holds, then Z_N^* is adaptive pseudo-free with respect to $\hat{\varphi}$*

Proof:- For the sake of contraction, we assume that RSA is not adaptive Pseudo-free with respect to multivariate equations. As per our definition of Adaptive pseudo-free groups with respect to multivariate equations there exists a probabilistic polynomial time adversary $\mathcal{A}$ with non-negligible probability that is supposed to produce an equation λ^* together with a solution ψ^* in such a way that λ^* is non-trivial with respect to previous queried equations.

For proof we construct an adversary $\mathcal{B}$ from an adversary $\mathcal{A}$ such that $\mathcal{B}'s$ main goal is to break strong RSA assumption. More specifically, adversary $\mathcal{B}$ is constructed in such a manner that if adversary $\mathcal{A}$ can break the adaptive pseuco-freeness of RSA group with respect to multivariate equations, then $\mathcal{B}$ can break strong RSA assumption

Let $\hat{\varphi}(M_k)\rightarrow (e_i^k, s_j^k, r_i^k)$ for all $k \in (1...t), i \in (1...n), j \in (1...m)$.
where 'k' is the number of queries done by adversary $\mathcal{A}$.
In order to prove the theorem we show two types of adverseries if we consider e_i^* from the equation(the adversary is supposed to produce) and e_i^k (already queried equations) for all $k \in (1...t), i \in (1...n)$.

TYPE 1-
Adversary produce e_i^* such that e_i^* doesnot divide $\prod_{i=1}^{n}\prod_{k=1}^{t} e_i^k$.
TYPE 2-
Adversary produce e_i^* such that e_i^* divides $\prod_{i=1}^{n}\prod_{k=1}^{t} e_i^k$.

Type 1 -
In this case we construct Adversary $\mathcal{B}$ that can break strong RSA assumption. As per Strong RSA assumption, it is not possible for an adversary given y and N to find x and e such that

$$x^e = ymodN$$

where $'N'$ is the product of two safe primes p and q and 'y' has been taken from QR_N. Thus Adversary $\mathcal{B}$ has input y and N and his aim is to compute e-th root x of y for some e of his own choice.

The following scenario consist of three stages as follows.

(1) Setup -

In this phase Adversary $\mathcal{B}$ randomly selects $\{r_1^1, ..., r_n^t\}$ and computes $\{e_1^1, ..., e_n^t\}$ as follows

$e_i^k = H(r_i^k)$ for all (i = 1,......n) and (k = 1,......,t).
Then $\mathcal{B}$ randomly selects $\{z_1, z_2, z_3, ..., z_m\}$ from $\{1, ..., N^2\}$ in order to set assign-

ment α for constant symbols.

Let E = $\prod_{i=1}^{n}\prod_{k=1}^{t} e_i^k$.

Thus $\mathcal{B}$ sets assignments as follows.
$\alpha(a_1) = y^{Ez_1}$
$\alpha(a_2) = \alpha(a_1)^{z_2}$
Thus $\alpha(a_j) = \alpha(a_1)^{z_j}$. for all ($j$ = 2,.....,.m).

$\mathcal{B}$ also randomly selects $\{z_1, z_2, z_3, ..., z_n\}$ from $\{1, ..., N^2\}$ and sets the constant variable symbols as follows :
$x_2 = y^{Ez_2}$
Thus $x_i = (x_2)^{Z_i}$ for i = 3 to n.

Finally Adversary $\mathcal{B}$ send assignments α and description of the computational group to adversary $\mathcal{A}$.
Let $Z_i = b_i p'q' + c_i$. is selected randmoly from $\{1, 2, ..., N^2\}$ and distribution of each and every ($z_i mod p$'q') is indistinguishable from uniform distribution over $\{Z_{p'q'}\}$.

(2) Equation Queries -

In this phase adversary $\mathcal{A}$ adaptively query equations by supplying parameters $\{M^1, ..., M^t\}$ for $\hat{\varphi}$. Thus adversary $\mathcal{B}$ is allowed to compute solutions of equations (for k = 1 to t) for adversary $\mathcal{A}$ and finally provides equations and their solutions to $\mathcal{A}$. In order to solve each equation $\mathcal{B}$ selects exponents according to parametric distribution $\hat{\varphi}(M_k)$ (for all k = 1 to t) from $\{s_2^k, s_3^k, ..., s_m^k\}$.
Therefore $\mathcal{B}$ solve equations as

$\lambda_i \equiv \{x_1^{e_1^k} x_2^{e_2^k} .....x_n^{e_n^k} = a_1 a_2^{s_2^k} .....a_m^{s_m^k}\}_{k=1}^{t}$

Let $E_i = \prod_{i=1}^{n}\prod_{j=1}^{t} e_i^j$ where $j \neq k$.
The above equations can be also written as

$$x_1^{e_1^k} = \frac{a_1 a_2^{s_2^k} ....a_m^{s_m^k}}{x_2^{e_2^k} ....x_n^{e_n^k}}$$

Thus by substituting the values of $\{a_1, a_2, ..., a_m\} and \{x_2, x_3, ..., x_n\}$ we have

$$\psi_k(x_1) = \frac{(y^{E_i})^{Z_1 + \sum_{j=2}^{m} s_j^k Z_j}}{(y^{E_i})^{Z_2 e_2^k + \sum_{i=3}^{n} e_i^k Z_j}}$$

$$= (y^{E_i})^{Z_1 + \sum_{j=2}^{m} s_j^k Z_j - \sum_{i=2}^{n} e_i^k Z_j}$$

Finally $\mathcal{B}$ gives $(e_i^k, s_j^k, r_i^k, \psi(x_i))$ to $\mathcal{A}$.

(3) Challange -

Once the adversary $\mathcal{A}$ has seen equations along with their solutions in the previous phase, then in this stage he tries to produce an equation λ^* along with its solution ψ^* such that λ^* is non-trivial to previously queried equations.
Hence (e_i^*, s_j^*, r_i^*) is distributed with respect to $\hat{\varphi}(M^*)$
Thus $\lambda^* \equiv \{x_1^{e_1^*} x_2^{e_2^*} ....x_n^{e_n^*} = a_1 a_2^{s_2^*} ....a_m^{s_m^*}\}$

$$\psi^*(x_1) = \frac{(y^E)^{Z_1 + \sum_{j=2}^m s_j^* Z_j}}{(y^E)^{Z_2 e_2^* + \sum_{j=3}^n e_i^* Z_j}}$$

$$= (y^E)^{Z_1 + \sum_{j=2}^m s_j^* Z_j - \sum_{j=2}^n e_i^* Z_j}$$

Let $\hat{\text{E}} = E(Z_1 + \sum_{j=2}^m s_j^* Z_j - \sum_{j=2}^n e_j^* Z_j)$ and u = GCD(e_i^*, $\hat{\text{E}}$). Provided e_i^* does not divide $\hat{\text{E}}$. Therefore adversary $\mathcal{B}$ can use shamir's trick to get (e_i^*/u)-th root of x of y for e of his own choice and thus give (e_i^*/u,x) to break strong RSA assumption.
Finally we need to show that (e_i^*) does not divide $\hat{\text{E}}$ with non-negligible probability. If we consider some prime r that divides (e_i^*). As we are in Type 1 adversary such that r does not divide E. Therefore we have to show that r does not divide $Z_1 + \sum_{j=2}^m s_j^* Z_j - \sum_{j=2}^n e_j^* Z_j$. As we know that $Z_i = b_i p' q' + c_i$.Since b_i is not known to adversary and r depends on c_i. Thus r does not divide *p'q'* and the probability with which $Z_1 + \sum_{j=2}^m s_j^* Z_j - \sum_{j=2}^n e_j^* Z_j = 0$ mod r is $1/r$. Thus we can say that the probability with which (e_i^*) does not divide $\hat{\text{E}}$ is (1-1/r) for small prime factor r of (e_i^*).

Type 2 -

In this case adversary outputs e_i^* such that e_i^* divides $\prod_{i=1}^n \prod_{k=1}^t e_i^k$. Thus we have two sub-cases as follows:

1) In this case for all $k = 1$ to t and $i = 1$ to n ,$(r_i^* \neq r_i^k)$. Therefore our assumption on $\hat{\varphi}$ is not satisfied as we would be able to break division intractability of the function $\mathbf{H}$. As we have $(r_1^1, ...r_n^t)$ and $(r_i^* \neq r_i^k)$ for all $k = 1$ to t and $i = 1$ to n such that $\mathbf{H}(r_i^*) = e_i^* \mid \prod_{i=1}^n \prod_{k=1}^t e_i^k$ where $\mathbf{H}(r_i^k) = (e_i^k)$.

2) In this case for some $j = 1$ to t and $i = 1$ to n ,$(r_i^* = r_i^j)$ where $j \neq k$. Therefore we have $(e_i^* = e_i^j)$. Now we give simulation for the given case below. First of all we describe how to construct an adversary $\mathcal{B}$ whose main goal is to break strong RSA assumption.

Before proceeding towards simulation we provide some details that will be usefull for reader to understand this approach.

Let (e_i^k, s_j^k) be the exponents of λ_k for all k = *1 to t* and (e_i^*, s_j^*) be the exponents of λ^* . since the equation λ^* is non-trivial, thus

$$\exists k \in Z^t, V \in Z^m such that : e^*(\sum E_r k + V_r) \neq s^*$$

.

We show the above condition in matrix form as:

$$e_i^*(\begin{bmatrix} 1 & \cdots & 1 \\ s_2^1 & \cdots & s_2^t \\ \vdots & & \vdots \\ s_m^1 & \cdots & s_m^t \end{bmatrix} \begin{bmatrix} k_{1,1}^1/e_1^1 & \cdots & k_{1,1}^n/e_1^t \\ \vdots & & \vdots \\ k_{t,n}^1/e_n^1 & \cdots & k_{t,n}^n/e_n^t \end{bmatrix} + \begin{bmatrix} v_1^1 & \cdots & v_1^n \\ v_2^1 & \cdots & v_2^n \\ \vdots & & \vdots \\ v_m^1 & \cdots & v_m^n \end{bmatrix}) \neq \begin{bmatrix} 1 \\ s_2^* \\ \vdots \\ s_m^* \end{bmatrix}$$

Thus the above comdition is true for all $k \in Z^t, V \in Z^m$, then it must be true for some $\hat{k}$ and $\hat{V}$. As we are in the case where $(e_i^* = e_i^j)$. It is easy to see that the first equation is always true for $\hat{k}$ and $\hat{V}$. Then it will also be true for $(s_v^j \neq s_v^*)$ for some v = 2 to *m*. Adversary $\mathcal{B}$ can randomly select j from (1 to t) and v from (2 to m) and then perform the simulation as follows:

(1) Setup -

In this phase Adversary $\mathcal{B}$ randomly selects $\{r_1^1, ..., r_n^t\}$ and computes $\{e_1^1, ..., e_n^t\}$ as follows

$e_i^k = H(r_i^k)$ for all (i = 1 to n) and (k = 1 to t).
Then $\mathcal{B}$ randomly selects $(g_1, g_2, ..., g_m)$ from QR_N and Z_v, ρ from $\{1, 2...., N^2\}$ and set assignments as:

Let E = $\prod_{i=1}^n \prod_{k=1}^t e_i^k$.

Thus $\mathcal{B}$ sets assignments as follows.
$\alpha(a_2) = y^E$
Thus $\alpha(a_v) = \alpha(a_2)^{z_v}$. for all ($v$ = 2,......,m).
$\alpha(a_1) = \alpha(a_2)^{-\rho} g_1^E$

Thus $\alpha(a_i) = (g_i)^{E_l}$ from i = (3,...,m) and $i \neq v$.
where $E_l = \prod_{i=1}^n \prod_{l=1}^t e_i^l$

$\mathcal{B}$ also randomly selects $\{z_1, z_2, z_3, ..., z_n\}$ from $\{1, ..., N^2\}$ and set the constant variable symbols as follows :
$x_2 = y^{Ez_2}$
Thus $x_i = (x_2)^{Z_i}$ for $x = 3$ to n.

Finally Adversary $\mathcal{B}$ sends assignments α and description of the computational Group to adversary $\mathcal{A}$.

(2) Equation Queries -

In this phase adversary $\mathcal{A}$ adaptively query equations by supplying parameters $\{M^1, ..., M^t\}$ for $\hat{\varphi}$. Thus adversary $\mathcal{B}$ is allowed to compute solutions of equations $(fork = 1tot)$ for adversary $\mathcal{A}$ and finally provides equations and their solutions to $\mathcal{A}$. In order to solve each equation $\mathcal{B}$ selects exponents according to parametric distribution $\hat{\varphi}(M_k)$ $(forallk = 1tot)$ from $\{s_2^k, s_3^k, ..., s_m^k\}$.
Therefore $\mathcal{B}$ solve equations as

$\lambda_i \equiv \{x_1^{e_1^k} x_2^{e_2^k} ....x_n^{e_n^k} = a_1 a_2^{s_2^k} ....a_m^{s_m^k}\}_{k=1}^t$
The above equations can be also written as

$$x_1^{e_1^k} = \frac{a_1 a_2^{s_2^k} ....a_m^{s_m^k}}{x_2^{e_2^k} ....x_n^{e_n^k}}$$

Thus by substituting the values of $\{a_1, a_2, ..., a_m\} and \{x_2, x_3, ..., x_n\}$ we have

$$\psi_k(x_1) = \frac{(y^{E_l})^{1+Z_v s_v^k - \rho(\prod_{j=1,j\neq 2,v}^m u_j^{s_k^j})^{E_l}}}{(y^{E_l})^{Z_2 e_2^k + \sum_{i=3}^n e_i^k Z_j}}$$

$$\psi_k(x_1) = (y^{E_l})^{1+Z_v s_v^k - \rho(\prod_{j=1,j\neq 2,v}^m u_j^{s_j^k})^{E_l} - \sum_{i=2}^n e_i^k Z_j}$$

Thus for solving j-th equation adversary $\mathcal{B}$ adopt a different technique as follows:
Let M^j be the queried parameter.In order to solve each equation $\mathcal{B}$ selects exponents according to parametric distribution $\hat{\varphi}(M^j)$ $(forallj = 1tot)$ from $\{s_2^j, s_3^j, ..., s_m^j\}$.

Then $\mathcal{B}$ fix $s_2^j = \rho - Z_v s_v^j mode_i^j$ and compute δ in such a way that $\rho - Z_v s_v^j = s_2^j + \delta e_i^j$.Then $\mathcal{B}$ then find

$$\psi_j(x_1) = \frac{(y^{-\delta})(\prod_{i=1,i\neq 2,v}^m u_i^{s_i^j})^{E_l}}{(y^{E_l})^{Z_2 e_2^k + \sum_{i=3}^n e_i^k Z_j}}$$

$$= \psi_j(x_1) = (y^{-\delta - E_l(\sum_{i=2}^n e_i^k Z_j})(\prod_{i=1,i\neq 2,v}^m u_i^{s_j^i})$$

$$= \sqrt[e_i^j]{(a_1 a_2^{s_2^j} \ldots\ldots\ldots\ldots a_m^{s_m^j} x_2^{-e_2^j} \ldots\ldots\ldots\ldots x_n^{-e_n^j})}$$

Finally $\mathcal{B}$ gives $(e_i^k, s_j^k, r_i^k, \psi(x_i))$ to $\mathcal{A}$ after solving each equation.

(3) Challange -

Once the adversary $\mathcal{A}$ has seen equations together with their solutions in the previous phase, then in this stage he tries to produce an equation λ^* along with its solution ψ^* such that λ^* is non-trivial to previously queried equations.
Therefore $\mathcal{B}$ will find root of y as follows:

We can write

$$(\frac{\psi^*(x_1)}{\psi_j(x_1)})^{e_i^*} = \frac{a_1 a_2^{s_2^*} \ldots\ldots a_m^{s_m^*} x_2^{-e_2^*} \ldots\ldots x_n^{-e_n^*}}{a_1 a_2^{s_2^j} \ldots\ldots a_m^{s_m^j} x_2^{-e_2^j} \ldots\ldots x_n^{-e_n^j}}$$

$$= a_2^{(s_2^* - s_2^j) + z_v(s_v^* - s_v^j)} (\prod_{i=1, i \neq v}^{m} a_i^{(s_j^* - s_i^j)})(\prod_{i=2}^{n} x_i^{-(e_i^* + e_i^j)})$$

$$= (y^{E_l})^{(s_2^* - s_2^j) + z_v(s_v^* - s_v^j)} (\prod_{i=1, i \neq v}^{m} u_i^{(s_i^* - s_i^j)})^{(E_l) e_i^j} (\prod_{i=2}^{n} x_i^{-(e_i^* + e_i^j)})$$

where $(l \neq j)$.

Hence we have a case where $e_i^* = e_i^j$, therefore by substituting the value of x_i for all (i = 2 to n) we get

$$[(\frac{\psi^*(x_1)}{\psi_j(x_1)})(\prod_{i=1, i \neq v}^{m} u_i^{(s_i^* - s_i^j)}))^{(E_l)}]^{e_i^*} = (y^{E_l})^{(s_2^* - s_2^j) + z_v(s_v^* - s_v^j) - (\sum_{i=2}^{n} -(e_i^* + e_i^j) Z_j)}$$

Let $\hat{E} = E_l((s_2^* - s_2^j) + z_v(s_v^* - s_v^j) - (\sum_{i=2}^{n} -(e_i^* + e_i^j) Z_j))$.Finally we need to show that (e_i^*) does not divide Ê with non-negligible probability. As we know that $Z_i = b_i p' q' + c_i$.Since b_i is not known to adversary. we can see that $(e_i^*) \mid E_l$ and also $s_v^* - s_v^j \neq 0$, we have that
$(e_i^*) \mid (s_2^* - s_2^j) + z_v(s_v^* - s_v^j) - (\sum_{i=2}^{n} -(e_i^* + e_i^j) Z_j)$ with negligible probability. Therefore adversary $\mathcal{B}$ can use shamir's trick to get (e_i^*/u)-th root of x of y to break strong RSA assumption where u = GCD(e_i^*, Ê).

The theorm RSA is adaptive pseudo-free with respect to multivariate equations can be proved by means of corollary using two new parametric distributions $(\hat{\varphi_S}, \varphi_{\hat{CH}} \neq \hat{\varphi})$ such that $\hat{\varphi_S}$ is a variant of $\hat{\varphi}$

Corollary 1 [9] :- *If the strong RSA assumption holds, then Z_N^* is adaptive pseudo-free with multivariate equations with respect to $\hat{\varphi_S}$.*

The proof for the given corollary follows from that of theorem 1. The main idea here is that one can guess s_i in advance with non-negligible probability when they are small. As we have defined above that $\hat{\varphi_S}$ is a variant of $\hat{\varphi}$ such that s_2 is equal to 0 and $s_i \forall (i = 3tom) \in Z_e$ is the output of chameleon hash function $CH(M, R)$ defined over parameter M and randomness R.

Corollary 2[9] :- *If the strong RSA assumption holds and CH is a chameleon hash function, then Z_N^* is adaptive pseudo-free with multivariate equations with respect to $\hat{\varphi_{CH}}$.*

The corollary has same proof as defined in corollary 1. The main idea here is that one can use the property of chameleon hash functions during simulation in order to choose s_i in advance.

Weak Adaptive Pseudo-freeness of the RSA Group:-

In weakest form of adaptive pseudo-free groups, the adversary at the start is allowed to select parameters $M^1, ..., M^t$ before giving any group description.If one can think of such a notion of adaptive pseudo-freeness then we can prove theorem 1 in more general way with respect to much general distribution than $\hat{\varphi}$ where $(e, s_2, ..., s_m)$ needs to be bound to M. Thus we can see one can compute all $r_i's$ in advance at the start because the simulator knows $M^1, ..., M^t$.

6 Pseudo-free Groups with System of Equations

6.1 Non-Trivial System of Equations

In this section we extend the notion of adaptive pseudo-free groups from multivariate to system of equations. The intuition behind this is to allow adversary to come up with a new set of equations. In definition [11] the adversary is allowed to see some non-trivial equations along with solutions and finally he has to come up with a new non-trivial equation and solution. Therefore in this case the game is same as the adversary is given a set of non-trivial multivariate equations but here he also has to come up with a new set of equations that are non-trivial to the given set. Formally we define as below:

Let

$$\Lambda = \{x_1^{e_1^k} x_2^{e_2^k} .....x_n^{e_n^k} = a_1^{s_1^k} a_2^{s_2^k} .....a_m^{s_m^k}\}_{k=1}^{t}$$

be a set of multivariate equations over the free abelian group 'F' and let $\{\phi_{1,k}, \phi_{2,k}, ..., \phi_{n,k}|$ κ = 1.....t$\}$ be a solutions for these equations. Assume the free abelian group 'F' is generated by $\{\phi_1, \phi_2, ..., \phi_n \ , \alpha_1, \alpha_2, ..., \alpha_n\}$

Definition 12 : *Set of Equations* $\Lambda' = \{x_1^{e_1^j} x_2^{e_2^j} ....x_n^{e_n^j} = a_1^{s_1^j} a_2^{s_2^j} ......a_m^{s_m^j}\}_{j=1}^{t}$ *is trivial with respect to* Λ *if there exists solution of the such system of equations over* $F/\!\equiv_\Lambda$

Let

$$\{x_1^{e_1^j} x_2^{e_2^j} .....x_n^{e_n^j} = a_1^{s_1^j} a_2^{s_2^j} ....a_m^{s_m^j}\}_{j=1}^{t} \tag{4}$$

be a system of equations

Assume that

$$\phi_i^j = \prod_{y=1}^{m} a_y^{v_y^i} \prod_{y=1}^{n} \prod_{l=1}^{t} \phi_{y,l}^{k_{l,y}^i}$$

is a solution for the equation for some $v_y^i, k_{l,y}^i$ for $(1 \le y \le m, 1 \le j \le k, 1 \le l \le n, 1 \le i \le n)$ and rationals $l_1, l_2....l_t$ for some $\mathbf{Q}$ such that:

$$\{(\phi_1^{k_1^1}, ..., \phi_1^{k_n^t}, a_1^{v_1^1}...a_1^{v_1^k})^{e_1^j}....(\phi_n^{k_1^1}, ..., \phi_n^{k_n^t}, a_m^{v_m^1}...a_m^{v_m^k})^{e_n^j}\} = a_1^{s_1^j} a_2^{s_2^j} .....a_m^{s_m^j} \prod_{i=1}^{t} (\prod_{y=1}^{n} \phi_{y,i}^{-e_y^i} \prod_{y=1}^{m} a_y^{s_y^i})^{l_i}$$

$$(\prod_{i=1}^{n} \phi_i^j)_{(j=1..t)}^{e_i^j} = a_1^{s_1^j} a_2^{s_2^j} .....a_m^{s_m^j} \prod_{i=1}^{t} (\prod_{y=1}^{n} \phi_{y,i}^{-e_y^i} \prod_{y=1}^{m} a_y^{s_y^i})^{l_i} \tag{5}$$

Thus by replacing the expressions in the above relation for ϕ_i^j and matching exponents of different symbols we have an equations:

$$\{x_1^{e_1^j} x_2^{e_2^j} .....x_n^{e_n^j} = a_1^{s_1^j} a_2^{s_2^j} ......a_m^{s_m^j}\}_{j=1}^{t} \tag{6}$$

is trivial with respect Λ to if there exists $v_y^i, k_{l,y}^i$ for $(1 \leq y \leq m, 1 \leq l \leq n, 1 \leq i \leq n)$ and rationals $l_1, l_2....l_t$ for some $\mathbf{Q}$ such that

1) $\sum_{i=1}^{n} k_{u,y}^i e_i^j = e_y^u l_u$ for $(1 \leq u \leq t, 1 \leq y \leq n)$

2) $\sum_{i=1}^{m} v_y^i e_i^j = s_y^j - \sum_{u=1}^{t} s_y^u l_u$ for $(1 \leq y \leq m)$

The converse of the above statement also true if there exists if there exists $v_y^i, k_{l,y}^i$ for $(1 \leq y \leq m, 1 \leq l \leq n, 1 \leq i \leq n)$ and rationals $l_1, l_2, ..., l_t$ for some $\mathbf{Q}$ such that equation (2) holds then

$$\phi_i^j = \prod_{y=1}^{m} a_y^{v_y^i} \prod_{y=1}^{n} \prod_{l=1}^{t} \phi_{y,l}^{k_{l,y}^i}$$

is a solution for the equation (1) over F/$\equiv_\Lambda$

6.2 Adaptive Pseudo-free Groups With System of Equations

The formal definition of adaptive pseudo-free groups[9] given below is for a set 'A' of α generators, a computational group $\{G_N\}_N$ and a distribution ϕ over set of equations .

Setup

In this phase first of all the challenger randomly selects instance of computational group $\{G_N\}$ that belongs to a family of $\mathcal{G}$ where $\mathcal{G} = \{G_N\}_{N \in Nk}$ and then sets an assignment n: A→ $\{G_N\}$ for set A of generators and parametric distribution φ. At last the input to the assignment n: A→ $\{G_N\}$,description of the computational Group $\{G_N\}$ and parametric distribution φ is given to the Adversary $\mathcal{A}$.

Equation queries

In this phase the adversary is permitted to look at equations together with their solutions. The queried equations are controlled by the adversary through parametric distribution φ. Thus the adversary $\mathcal{A}$ chooses $\mathcal{M}_k$ for each query and gives it to challenger. Then the challenger runs $(e_i^k, s_j^k) \leftarrow \hat{\varphi}(\mathcal{M}_k)$, computes the solution ψ_i of equation λ_i and gives back $(\psi_i$,$(e_i^k, s_j^k))$ to adversary $\mathcal{A}$.

Challenge

After the adversary is acquaimted with equations and their solutions, then he tries to produce set of equations λ_j with a solution ψ_j for(j= 1..t).The adversary succeeds in the game if he successfully produces a non trivial equation.

Definition 13 [9] Adaptive Pseudo-free Groups with System of Equations:- *A family of computational Group $\mathcal{G}$ is said to be adaptive Pseudo-free with respect to some distribution φ, if for any set 'A' of polynomial size, any probabilistic time adversary $\mathcal{A}$ succeeds in the game above with at most negligible probability.*

7 Conclusion

In this project we have seen that the concept of pseudo-freeness seems to be a strong, natural and quite plausible in commonly used groups in cryptography. We have explored the notion of adaptive pseudo-free groups from univariate to multivariate and system of equations. We have studied and identified pseudo-free group with respect to adaptive adversaries who aim to solve non trivial equations along with their solutions. First of all we have shown triviality for the case of adaptive pseudo-free groups and then introduced formal definition of such groups for multivariate and system of equations.

There are many open problems left by Rivest[31]. As one of the open problem is to find groups in cryptography that satisfies the notion of pseudo-free groups. Thus for a case we have proved the adaptive pseudo-freeness of the RSA group (Z_N^*) with multivarite equations under reasonable conditions such that N is the product of two safe primes. Another problem in [31]is to prove the pseudo-freeness of Diffe-Hellman assumption as we have seen that the Diffe-Hellman assumption(either computational or decisional) does not satisfies the property of pseudo-free groups. In this project we have not addressed this problem but in future it would be interesting to see whether Diffe-Hellman assumption holds our definition of adaptive pseudo-free groups.

Another open research in this direction is to to find crytographic applications[26] that provides the usefullness of such groups in cryptography. In next section we have just provide an idea of extracting an application out of adaptive pseudo-free groups with respect to multivarite equations. It would be interesting to see whether that application really holds our definition and can capture use of pseudo-free groups in cryptography.

8 Future Work

The study of pseudo-free groups carried out so far did not allow us to extract applications out of them because of prime generation. As an first application of adaptive pseudo-free groups (a signature scheme) for univariate equations defined in [9], is more efficient than the current RSA based signature schemes but it requires a longer public key. It may be a case that we can resolve this issue by constructing a signature scheme out of adaptive pseudo-free groups with multivariate equations. One way to construct a more efficient signature scheme is to consider equations that do not have prime exponents as prime generation is the bottleneck.

The cryptographic assumptions such as discrete logarithm problem and the Diffie-Hellman problem[26] are considered at least as hard as factoring and currently there is no relation between pseudo-free groups and these assumptions. An interesting work is to see that whether such cryptographic assumptions in pseudo-free groups are computationally hard. Finally it would be interesting to find relationship between pseudo-freeness and other cryptographic assumptions with the intuition of using such groups in cryptography.

9 References

[1] M.Abadi and P.Rogaway, Reconciling two views of cryptography(the computational soundness of formal encryption). *Journel of Cryptology*, 20(3):395, July 2007.

[2] L.Babai. Randomization in group algorithms:conceptual questions. In L.Finkelstein and W.M.Kantor,editors,*Groups and Computation 2. Proc 1995 DIMACS Workshop,volume 28 of DIAMS Ser.in Discr.Math and Theor.Comp.Sci.,*pages 1-16.AMS,1997.

[3] E.Bach, Discrete logrithms and factoring.Technical Report CSD-84-186,University of Californiaat Berkley(1984).

[4] Niko Baric and Birgit Pfitzmann. Collision-free accumlators and fail stop signature schemes without trees. In *Proc EUROCRYPT '97*, volume 1233 of *Lecture notes in Computer Science*, pages 480-494. Springer-Verlag,1997.

[5] Mihir Bellare and Phillip Rogway. Random oracles are practical: A paradigm for desiging efficient protocols. *First ACM Conference on Computer and Communications Security*, pages 62-73, Fairfax, Virginia, USA, November 3-5, 1993. ACM Press.

[6] Dan Boneh and Richard J.Lipton. Algorithms for black box fields and their application to cryptography. *Advances in Cryptology*:283-297, 1996.

[7] Christian Cachin,Silvio Micali and Markus Stadler. Computationally private information retrival with polylogarithmic communication. In Jacques Stern, editor, *EUROCRYPT'99*, VOLUME 1592 OF *LNCS*, pages 402-414, Prague, Czech Republic, May 2-6, 1999. Springer, Berlin, Germany.

[8] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In stelvio Cimato, Clemente Galdi and Giuseppe Persiano,editors, *SCN 02*, volume 2576 of *LNCS*, pages 268-289, Amalfi, Italy, September 12-13, 2002. Springer,Berlin, Germany.

[9] Dario Catano, Dario Fiore and Bogdan Warinschi, Adaptive Pseudo-Free Groups and Applications. *Advances in cryptology*, pages 207-223, May 2011.

[10] Henri Cohen. *A course in Computational Algebraic Number theory.* Springer, 1993

[11] Leo P.Comerford,Jr and Charles C.Edmunds. Quadratic parametric equations over free groups. In K.I.Appel,J.G.Ratcliffe, and P.E Schupp, editors, *Contributions to Group theory, volume 33 of Contemparary Mathematics*, pages 159-196 AMS, 1984.

[12] R. Cramer and V.shoup, Signature schemes based on the strong RSA assumption. *ACM Trans.Inf.Syst.Secur.*3(3), 161-185 (2000). Preliminary version in CCS'99.

[13] W.Diffie and M.E.Hellman. Multiuser cryptographic techniques. In *Proc.AFIPS 1976 National Computer Conference*, pages 109-112, Montvale, N.J., 1976. AFIPS.

[14] W.Diffie and M.E.Hellman. New directions in cryptography. *IEEE Trans.Inform. Theory*, IT-22:644-654, November 1976.

[15] D.Dolev and A.C.Yao. On the security of public key protocols. In *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, pages 350-357, 1981.

[16] Claudio Gutierrez. Satisfiability of equations in free groups is in PSPACE. *In Proc.32nd ACM STOC*, pages 21-27, ACM Press, 2000.

[17] Susan Hohenberger, David Molnar and Ronald L.Rivest. Special signatures need special algebra, May 2003.Submitted.

[18] Susan Hohenberger. The cryptographic impact of groups with infeasible inversion. Master's thesis, EECS Dept., MIT,Cambridge, MA June 2003.

[19] Takato Hrano and Keisuke Tanaka. Variations on Pseudo-free groups. A research report. In department of mathematical and computing science in Tokyo institute of technology, Japan 2007.

[20] Olga Kharalampovich and Alexei Myasnikov. Implicit function theorem over free groups. Available at www.math.mcgill.ca/olga/publications.html.

[21] Olga Kharalampovich and Alexei Myasnikov. Tarski's problem about the elementry theory of free groups has a positive solution. *Electronic Research Announcements of the American Mathematical Society*, 4:101-108, December 14, 1998.

[22] Seymour Lipschutz and Charles F.Miller. Groups with certain Solvable and unsolvable Decision problems. *Communications on Pure and Applied Mathematics*, 7-15, 1971.

[23] R.C.Lyndon Equations in Free groups. *Trans. Amer. math.Soc.*,96:445-457, 1960.

[24] G.S.Makanin. Equations in a free group. *IZvestiya NA SSSR*, 46:1199-1273,1982. English translation in Math USSR Izvestiya, 21 (1983),483-556.

[25] Ueli.Maurer and Stefan Wolf. The relationship between breaking the Diffie-Hellman

protocol and computing discrete logrithms. *SIAM journel on Computing* 28(5): 1689-1721,1999.

[26] D. Micciancio. The RSA group is Pseudo-free. In Ronald Cramer,editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 387-403, Aarhus, Denmark, May 22-26,2005. Springer, Berlin, Germany.

[27] D.Micciancio.,S.Goldwasser.*Complexity of lattice Problems: a Cryptographic Perspective,* The Kluwer International series in Engineering and Computer Science, Volume 691. (Kluwer Academic,Bosten, 2002).

[28] D. Micciancio and S.Panjiwani, Adaptive security of Symbolic Encryption. In. *Theory of Cryptographic Conference-Proceedings of* TCC. LNCS, vol.3378, pp. 169-187(2005).

[29] D. Micciancio and Bogdan Warinschi, Soundness of formal encryption in presence of adaptive adversaries. In: *Theory of Cryptographic Conference- Proceedings of TCC'04*. LNCS, vol.2951 pp.133-151 (2004).

[30] A.A.Razborov. On system of equations in a free groups.*IZvestiya NA SSSR*, 48:779-832 (In Russian), 1984.English translation in Math. USSR IZvestiya 25,1(1985) 115-162.

[31] R.L.Rivest. On the notion of pseudo-free groups. In: *Theory of Cryptographic Conference- Proceedings of TCC'04*. LNCS,vol. 2951,pp. 505-521(2004).

[32] Joseph H.Silverman and joe Suzuki. Elliptic curve discrete logarithms and the index calcus. In $Proc.Asiacrypt'$98,volume 1514 of $LectureNotesinComputerScience$, Pages 110-125.Springer-verlag,1998.

[33] Hasegawa Shingo, Isobe Shuji, Shizuva Hiroki and Tashiro Katsuhiro. On the Pseudo-freeness and CDH assumption. Springer Verlag, 2009.

[34] Joseph H.Silverman and Joe Suzuki. Elliptic curve discrete logrithms and the index calcus. In $Proc.Asiacrypt'$98, volume 1514 of $lecturenotesIncomputerscience$, pages 110-125, Springer-Verlag, 1998.

[35] Huafei Zhu. New digital signature scheme attaining immunity to adaptive chosen-message attack. $ChinesejournalofElectronics$, 10(4): 484-486, October 2001.

[36] Huafei Zhu. A formal proof of Zhu's signature scheme. Cryptlogy ePrint Archieve. Report 2003/155,2003.http://eprint.iacr.org/.