A New Key Schedule Proposal
And
Anonymous Authentication in Vehicular Ad-hoc Networks (VANET)

Thesis Submitted In Partial Fulfillment of the Requirements
For The Degree Of

Masters of Technology
In
Computer Science and Engineering

By
Nitin Bansal (03CS3015)

Under the guidance of
Prof. Dipanwita Roychowdhury

" YOGA KARMASU
KAUSALAM "

Department of Computer Science and Engineering
Indian Institute of Technology
Kharagpur
May 2008

INDIAN INSTITUTE OF TECHNOLOGY
KHARAGPUR

CERTIFICATE

This is to certify that thesis titled "**A New Key Schedule Proposal and Anonymous Authentication in Vehicular Ad-hoc Networks**" submitted by **Nitin Bansal** (03CS3015) to the Department of Computer Science and Engineering is a bonafide record of work carried out under my supervision and guidance .The thesis has fulfilled all the requirements as per as regulation of this institute and is valid for submission and the evaluation purposes.

Prof. D. Roychowdhury
Dept. of Computer Science and Engg.
Indian Institute of Technology
Kharagpur -721302, India
May 2008

Acknowledgements

I would like to take this opportunity to express my sincere gratitude, respect and regards for **Prof. Dipanwita Roy Chowdhury** under whose guidance, constant encouragement, patience and trust, I have worked on this project. She exposed me to the research topic through proper counsel rigorous discussion and always showed great interest in providing timely support and suitable suggestion. I am thankful to her for the constructive criticism and suggestions for improvements in various stages of this work.

I would like to thank all the faculty members, laboratory staff for their cooperation and support.

I owe my thanks to my friend Umang Jain for his help throughout the project and the nice time we had working together. I am also thankful to my classmates for their company for five years and all friends who have directly or indirectly assisted me in my endeavors.

I am indebted to my parents and my sister for their love, support and inspiration throughout my life.

Nitin Bansal

Abstract

In this work, we analyze the AES key schedule; discuss its security properties and weaknesses that assist the execution of effective attacks. We then propose and analyze a more efficient key schedule making use of features and properties provided by linear and non-linear Cellular Automata (CA). CA has been shown to be capable of generating complex and random patterns out of simple rules. Therefore, it has been used to provide randomness and nonlinearity to the key schedule proposal.

This work also proposes a secure group communication scheme well suited to the environment of VANET. We analyze major security requirements needed for secure communication in VANET along with different types of attacks possible. We then propose a secure communication scheme based on Chinese Remainder Theorem and compare its performance with present schemes.

Contents

** Work of Chapter 3, 4, 5 has been done along with Umang Jain (03CS3005)*

Chapter 1

Introduction to Cellular Automata

1.1 Preliminaries on Cellular Automata

A one-dimensional cellular automaton consists of two things: a row of "cells" and a set of "rules". Each of the cells can be in one of several "states". The number of possible states depends on the automaton. In a two-state automaton, each of the cells can be 1 or 0.Over time, the cells can change from state to state. The cellular automaton's **rules** determine how the states change. When the time comes for the cells to change state, each cell looks around and gathers information on its neighbors' states. (Exactly which cells are considered “neighbors" is also something that depends on the particular CA.) Based on its own state, its neighbors' states, and **the rules of the CA**, the cell decides what its new state should be. All the cells change state at the same time

For example, in a 1-dimensional cellular automaton, the neighborhood of a cell x_i^t—where t is the time step (vertical), and i is the index (horizontal) in one generation—is $\{x_{i-1}^{t-1}, x_i^{t-1}, x_{i+1}^{t-1}\}$. There will obviously be problems when a neighborhood on a left border references its upper left cell, which is not in the cellular space, as part of its neighbor

Rule of a CA:

If the next state function of a cell is expressed in the form of a truth table, then the decimal equivalent of the output is called the rule number of the cellular automaton.

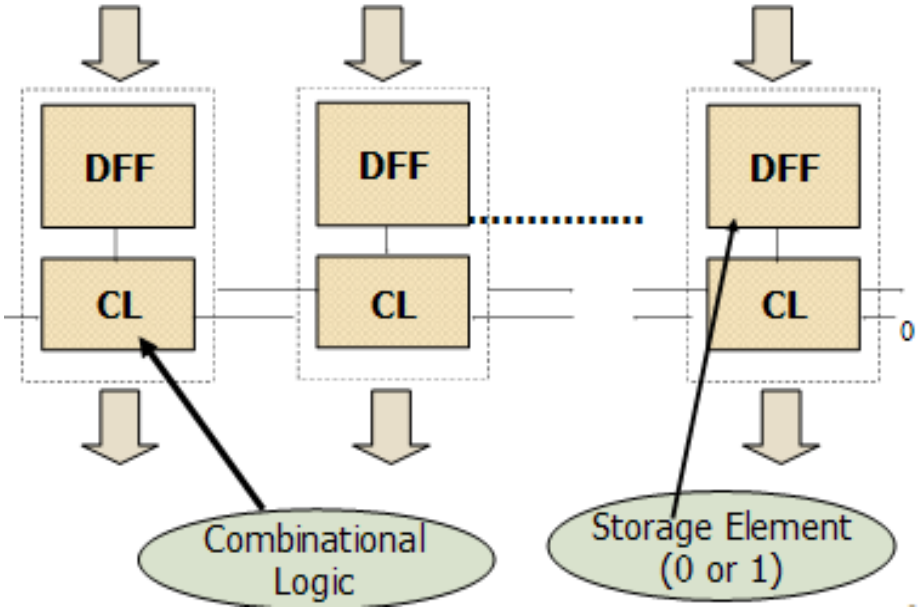


Fig 1.1: Diagram of Cellular Automata

Additive and Non Additive Cellular Automata [5]:

If the rule of a CA cell involves only XOR logic ,then it is called a linear rule .A CA with all the cells having linear rule is called a linear CA. Rules involving XNOR logic are referred to as complemented rules. A CA having a combination of XOR and XNOR rules is called an additive CA. *The rules with AND-OR logic are non-additive rules or non-linear rules.*

Uniform and Hybrid Cellular Automata:

If all the CA cells obey the same rule then the CA is said to be uniform CA, otherwise it is a hybrid CA

Null boundary CA:

A CA is said to be null boundary CA if the left (right) neighbor of the leftmost (rightmost) terminal cell is connected to logic 0 state

Periodic Boundary CA:

A CA is said to be Periodic Boundary CA if the extreme cells are adjacent to each other

Reversible CA:

A CA is said to be *reversible* if for every current configuration of the CA there is exactly one past configuration (preimage). If one thinks of a CA as a function mapping configurations to configurations, reversibility implies that this function is bijective.

Chapter 2

A New Key Schedule Proposal

2.1 Motivation

The Advanced Encryption Standard (AES) is the most significant standard of the block ciphers, so its security is of paramount importance. However, the key schedule of AES has a clear weakness that directly assists the execution of most effective attacks. To combat these weaknesses, we propose a different approach to the AES Key Schedule design. We demonstrate that it avoids the weakness of the existing key schedule.

The analysis of weak key schedules has led to the guidelines for robust key schedule design that borrows from well known and accepted design principles for block algorithms in the broader sense. Our design follows these key schedule guidelines.

2.2 Block Cipher Key schedules

The goal of a strong key schedules is to overcome any perceived weakness which may be used in attacking the block cipher system .Designers already ensure Shannon's property of confusion and diffusion properties in their cipher algorithms, so similar properties could be achieved for key schedules algorithms.

Biham [4] showed that in some simple cases, simple key schedules exhibit relationships between keys that may be exploited. Also Knudsen[6] listed four necessary but not sufficient properties for secure Fiestel ciphers .Two of these *,no simple*

relation and all keys are equal good ,are achievable with strong key schedules .The generic properties of a strong key schedule that are readily measurable are:

1): Function should be infeasible (or at least hard) to invert

2): Minimal mutual information (between all sub key bits and master key bits)

Property 1 ensures that given any round sub key it should be infeasible to get back the other round sub keys or master key just by inverting the functions used to get it.

Property 2 aims to eliminate bit leakage between sub keys and master keys, weakness that assists cryptanalysis by reducing the complexity of some attack scenarios on block ciphers. As some of the attacks make use of the relations between key bytes and would have a higher complexity if these relations did not exist.

Leakage of information from subkeys i to subkey i-1 or subkey i+1 is directly prevented by Property 2. Using master keys directly in subkeys leads to the worst case of bit leakage; however this can be easily avoided

2.3 AES Key Schedule

AES encryption algorithm is an iterative process where each of the rounds consists of nonlinear substitution, a linear transformation and a subkey addition. The schedule generates the round subkeys from the master keys .Possible weakness in the cipher introduced through this key schedule are highlighted.

2.3.1 Description of the key schedule

The key schedule is required to produce round subkeys from master keys. The schedule is based on 32 bit words .The initial words are set to equal the master key. The

remainder of the words is generated by an iterative process. Consecutive groups of four 32 bit words are concatenated to produce the 128 bit subkeys[2].

```
for i = 0 to 3
 W[i] = MasterKey[i]
for j = 4 to 40 (in steps of 4)
 W[j] = W[j-4]⊕ SubByte(Rotl(W[j-1]))⊕ Rcon[j/4]
 for i = 1 to 3
  W[i+j] = W[i+j-4]⊕ W[i+j-1]
```

2.3.2 Analysis

Several attacks have been mentioned in [8]which uses the weak key schedule to crypt analyze AES .The overriding security concern with the AES Key schedule ,therefore , is the fact that *,given knowledge of a round subkey (or part of a round subkey).knowledge of the other round subkeys (or parts) is immediately derivable*.

We now explicitly define this key schedule bit leakage problem as prelude to proposing a rectification. From the key schedule algorithm, it is noted that successive W[i] values are related to previous W[i] values. An example of this for the 128 bit key schedule is that knowledge of W[38] (32 bits of the Round 10 subkey).This is achievable since W[42]=W[38] xor W[41], and hence W[38] is explicitly determined by evaluating W[42] xor W[41] .It is noted that *every master key bit is not involved with the generation of the subkey bits until W[6].* The iterative nature of the key schedule is generally to enhance implementation efficiency, the problem, however, lies with the *definition of the iteration itself being too simplistic* which leads to the bit leakage problem.

Having defined the problem we wish to avoid in our new key schedule proposal, we outline our approach to the new design.

2.4 A New AES key schedule proposal

The analysis in the previous section highlights the fact that AES key schedule does not satisfy desirable properties outlined for key schedule .The aim of this section is to define a suitable key schedule which satisfies the desired properties .

Cellular Automaton (CA) has been shown to be capable of generating complex and random patterns out of simple rules. Moreover, they can be implemented efficiently in hardware .So it seems logical to include these in our key schedule design.

2.4.1 128 bit key schedule Proposal

Proposal 1

```
// First we create Rconstant for every round of AES
  For r = 0 to 10 {
      For j = 0 to 15 {
          rconstant j  = r* 16 + j
      }
  }
```

- Rconstant is of 128 bits and equal to $rconstant_0 | rconstant_1 | \ldots . rconstant_{15}$

  This Rconstant is different for every round and while generating round subkey we will use round constant corresponding to that round.

  (Here $rconstant_j$ is of 8bits and | represents concatenation)

- The inclusion of round dependent round constant (Rconstant) eliminates the symmetry, or similarity, between the ways in which round keys are generated in different rounds. It not only isolates each resulting subkey from others, but also breaks up possible weak keys, for example, if all the master keys were identical.

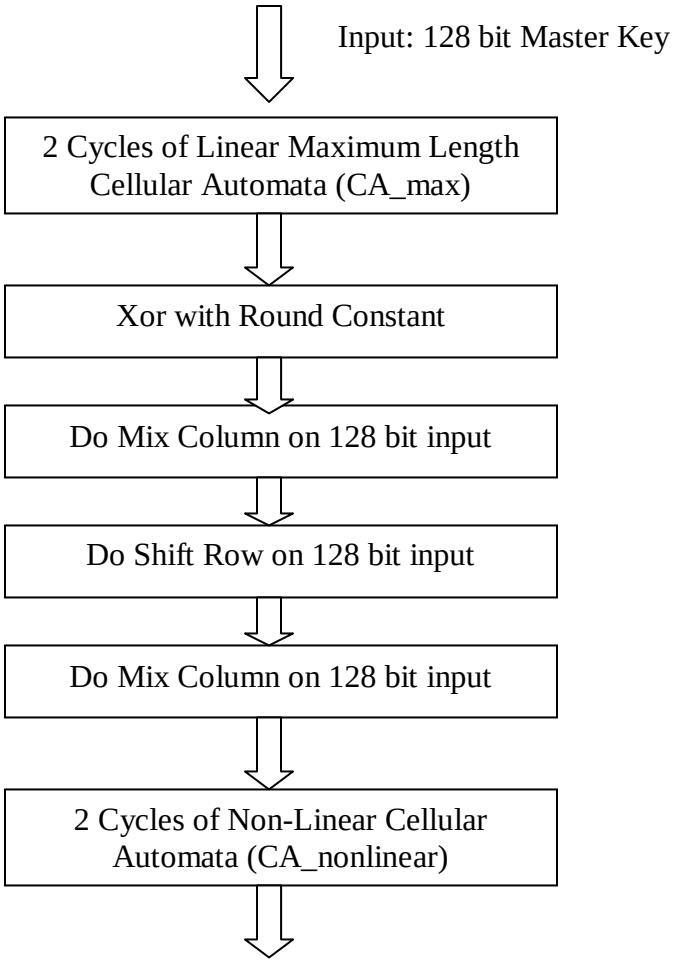


Fig 3.1: Key schedule proposal for 128 bit keys

- Final round key bits are selected on the basis of Master key

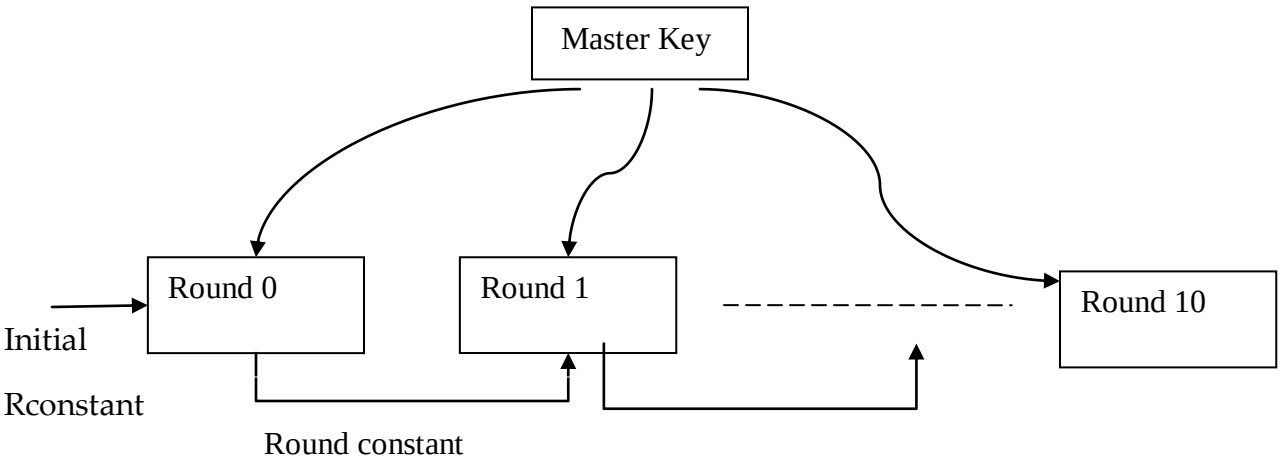
```
If Master key[i] = 1 then
    Subkey [i] =CA_nonlinear_output [cycle1] [i]
Else
        Subkey [i] = CA_nonlinear_output [cycle2] [i]
```

- CA_max uses 2 clock cycles of maximum length 128 bit CA where rules used are 90 and 150. We know that maximum length CA is random in nature. So this is used to provide randomness to the round sub keys.

- Mix column and shift row are the same operation used in AES round function and they are used to provide the required diffusion.

- CA_nonlinear is a 128 bit periodic nonlinear CA using rule 30. This is used to provide nonlinearity to the key schedule algorithm.

Proposal 2



In this proposal , round key of previous round acts as a round constant for the current round .However, the algorithm to generate round keys remain the same as one used in earlier proposal.

2.4.2 Hard to reverse

In our proposal we are not using the sub keys bit directly generated from Cellular automata using nonlinear rule 30 as it can be inverted with in linear time ie .if we know the complete successor state we can get the possible set of predecessors that generated that successor state with in linear time[10] .Thus to make this hard to reverse, we make the selection of subkeys bits based on the master key bits and these bits are selected from the outputs of CA_nonlinear() which is run for 2 cycles.

```
If Master key[i] = 1 then
    Subkey [i] =CA_nonlinear_output [cycle1] [i]
Else
        Subkey [i] = CA_nonlinear_output [cycle2] [i]
```

So even if cryptanalyst knows any particular subkey he will be not able to know from which CA output this particular bit was selected as he doesn't have master key with him. So every bit has 2 choices and which gives **2^128** cases to be considered and hence hard to reverse without the knowledge of master key. Thus this proposal overcomes the weakness of AES key schedule which can be inverted.

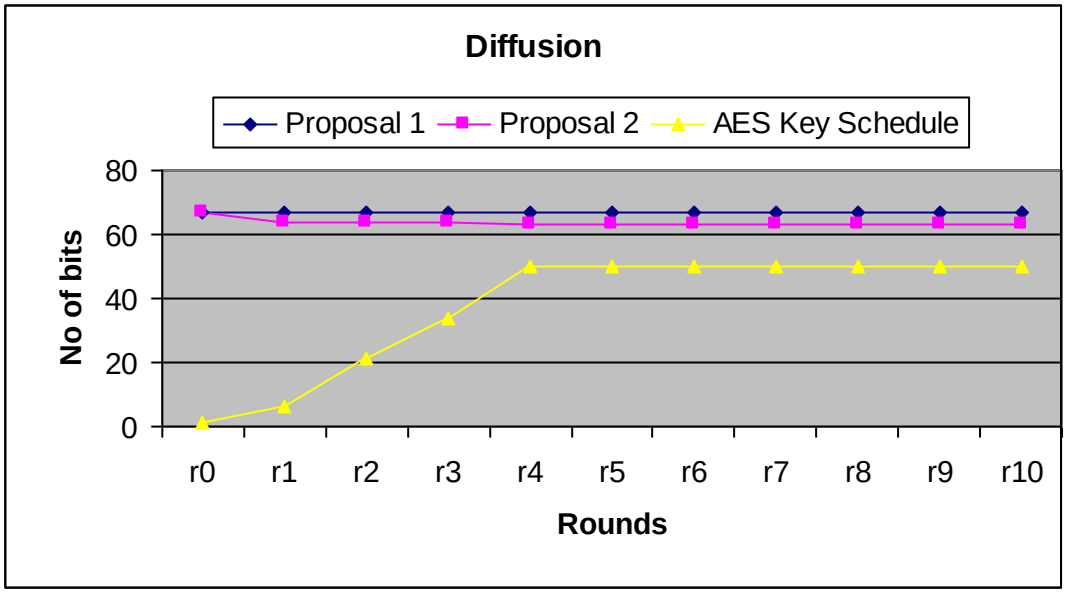
2.4.3 Diffusion properties

Tests were formed to calculate the number of bit changes in the output with a single bit change in the input. We found that for both proposals with a single bit change in input**, on an average half of the output bit changes which is a good measure of the Shannon's diffusion property [2].**

Moreover, complete diffusion was achieved when we used 2 clocks of the Cellular automata used and 2 rounds of Mix columns, thus clarifying the decision for choosing 2 CA clock cycles and 2 mix columns in the proposed key schedule.

Results were calculated over random input set of 1000 keys.

Rounds	r0	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
Proposal 1	67	67	67	67	67	67	67	67	67	67	67
Proposal 2	67	64	64	64	63	63	63	63	63	63	63
AES	1	6	21	34	50	50	50	50	50	50	50



2.4.4 Sub key Bit Difference

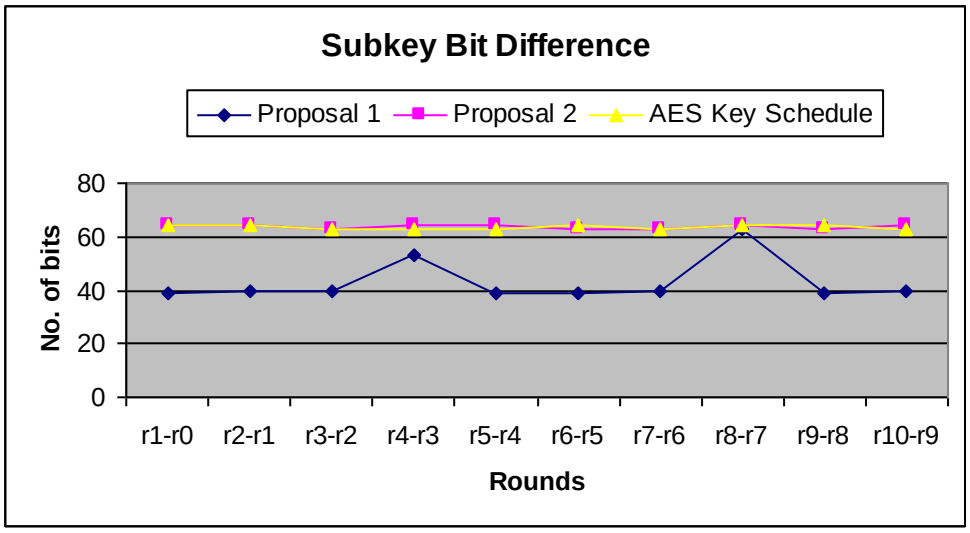
In order to build strong key schedule algorithm, for a given master key, algorithm should produce subkeys that differ in maximum number of bits so that cipher is provided with round keys that differ in maximum bits thereby making the block cipher strong.

So we calculated the difference between the consecutive subkeys over different randomly generated input keys (1000 random inputs were considered)

Rounds	r1-r0	r2-r1	r3-r2	r4-r3	r5-r4	r6-r5	r7-r6	r8-r7	r9-r8	r10-r9
Proposal 1	39	40	40	53	39	39	40	63	39	40
Proposal 2	64	64	63	64	64	63	63	64	63	64
AES	64	64	63	63	63	64	63	64	64	63

As we can see , proposal 1 doesn't really give good round keys whereas proposal 2 give round keys with difference close to 64 ie . Half of the bits differ between every consecutive subkeys. This property along with maximum diffusion helps in preventing related key attacks .A necessary condition for resistance against related-key attacks is that there should not be two different Cipher Keys that have a large set of Round Keys in common[7].

Thus key schedule with good diffusion and subkey difference prevents these attacks.



2.4.5 Bit variance test

The bit variance test consists of measuring the impact on the output bits of changing input messages .More specifically, given an input message, all the small changes as well as the large changes of this input message bits occur and the bits in the corresponding output are evaluated for each such change. Afterwards, for each digest bit the probabilities of taking on the values of 0 and 1 are measured considering all the digest produced by applying input messages bit changes. If P (0) =P (1) =0.5 for all the output bits, then, the function under consideration has attained maximum performance in terms of bit variance test [9].

Bit variance test actually measures the uniformity of each bit of the output. Since it is difficult to consider all the input message bit changes, we have evaluated the results for only up to two input message bit mutations.

More formally, for each digest bit, the bit variance test is defined as follows.

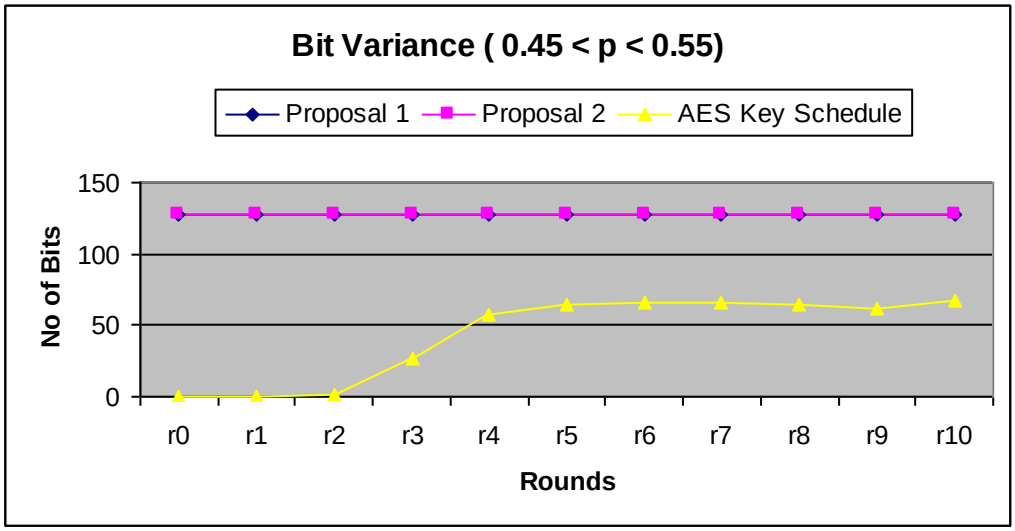
Let us assume a Boolean function f: F^n_2 -> F_2 where F is the set of all Boolean vectors with length n and F is the set {0, 1}. The Boolean function f is considered as passing the bit variance test if it satisfies the propagation criterion of degree k that is

(For all a: $1<W_H(a) < k$) P (f(X) = f(X ^ a))=0.5

Where, X is the input message, a is Boolean vector of length n and hamming weight $W_H(a)$ P defines probability and the symbol ^ define the Xor operation between Boolean vectors.

So we considered one random input key and consider 2000 mutations (1 bit and 2 bit mutation) of this key and calculated the number of output bits for whom the probability lies between 0.45 and 0.55.

Rounds	r0	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
Proposal 1	128	128	128	128	128	128	128	128	127	127	127
Proposal 2	128	128	128	128	128	128	128	128	128	128	128
AES	0	0	2	27	57	64	66	66	64	62	67



Greater the number of bits close to p=0.5 the more difficult it becomes to get back the input information from the output bits and test is generally used for checking one way hash functions properties.

So in our case AES key schedule shows poor performance in terms of bit variance test implying it will be easy to get back the information of input bit seeing the output or input information is not getting fully divulged with in the output bits whereas proposal 1 and proposal 2 shows considerably good performance and makes it difficult to retrieve input bits.

2.4.6 Security Analysis

- As each round sub key is generated independently in the proposal, and, consecutive sub keys differ in half of bits there is no bit leakage. Also the master key is not directly used as sub key in the proposal.

- **One wayness** is achieved by using master key bits in the selection of sub key bits from the outputs of nonlinear periodic CA having rule 30.

- High bit diffusion of each master key bit across each subkey is attained. This is particularly useful in thwarting related key attacks, as altering even one bit in the master key changes approximately half the bits in each subkey.

- A generic attack solicits some round subkey bits by forceful means. In contrast to the current AES key schedule, even if an entire 128 round subkey is known, as proven, it is hard to retrieve the master. It is not possible to obtain subkey bits from one round using material purely from another.

We believe the proposed key schedule to be safe from conventional methods of cryptanalysis.

2.5 Conclusion

We described and analyzed the AES key schedule in detail, and provided an overview of weakness in key schedule proposal. We presented and analyzed a new key schedule which adheres to basic Shannon's property of confusion and diffusion and proved it to be secure.

Chapter 3

Introduction to VANET

3.1 Vehicular Ad-Hoc Networks: An introduction

Vehicular Ad-Hoc Network, or **VANET**, is a form of Mobile ad-hoc network, to provide communications among nearby vehicles and between vehicles and nearby fixed equipment, usually described as roadside equipment. It uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars to connect and, in turn, create a network with a wide range. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is created.

The main goal of VANET is providing safety and comfort for passengers. To this end a special electronic device will be placed inside each vehicle which will provide Ad-Hoc Network connectivity for the passengers. This network tends to operate without any infra-structure or legacy client and server communication. Each vehicle equipped with VANET device will be a node in the Ad-Hoc network and can receive and relay others messages through the wireless network. Collision warning, road sign alarms and in-place traffic view will give the driver essential tools to decide the best path along the way.

VANET differ from Mobile Ad-Hoc Networks in some details. Rather than moving at random, vehicles tend to move in an organized fashion. The interactions with roadside equipment can likewise be characterized fairly accurately. And finally, most vehicles are restricted in their range of motion, for example by being constrained to follow a paved highway.

Once VANET is deployed successfully, it is set to revolutionize the way one looks at vehicles. Though in a way it will only extend the current trend of increasing automation in cars. Certain vehicles today already have modern technologies like Global Position System sensors or receivers and VANET is set to drastically increase the environment awareness of vehicles.

There are tremendous benefits to be reaped through the introduction of inter-vehicular communications. Advantages range from increased comfort and entertainment to enhanced safety and better organized traffic scenarios. But is also raises several issues in conceptualization as well as implementation thus giving ample research opportunities. Though at first the concerns regarding inter-vehicular communication may seem similar to those in any network, but the expected amount of data transmission, the huge number of vehicles and the relevance of geographical location of nodes make it much more challenging .Huge amounts of intellectual and monetary capital is being put in around the world to make VANET a reality.

Lot of work and consensus has already been established as far as the setting-up of network protocols are concerned. But at the heart of the VANET lies the communication protocols and therein arises the issue of securing the communications. This area till now has been under-explored and not until recently, have researchers started to pay more and more attention towards it. The problem of information security in VANET poses a different sort of challenge altogether. With the cars being expected to have limited storage and computational capabilities on board, it renders most of the standard algorithms impractical. Hence there is need to view security in VANET from a different perspective altogether.

In the coming sections we will describe the application areas of VANET and the security requirements for each of them followed by the state of the art in the field. In the coming chapters we put forth a proposal for security protocol for VANET.

3.2 Inter-Vehicular Communication: Applications

VANET is envisioned to have a varied range of applications once it is deployed in its full capacity, but it is expected that the full capacity will be achieved in due course of time, hence some applications have been deemed to be of a greater priority than others. It means that the initial focus must be on these applications. Keeping these in mind, the applications can be divided into two broad categories-:

3.2.1 Safety Applications: These applications refer to communication of information required for safety of the vehicles and the travelers. These include collision avoidance, using aggregated positioning and velocity information to ensure better traffic scenarios, fixing liabilities in case of accidents etc. Real life example may include situations like a vehicle transmitting message to inform others about accidents, landslides etc to prevent jams, notification of a road hazard or a road feature condition, warning about potential collisions and so on. These applications not only aim at preventing dangerous situations but also aim at identifying the culprits in case such a situation has occurred to help the law enforcement agencies. The security in these cases is paramount as false information may be propagated in the network for personal gains and of course the world is not devoid of cynical people who would aim to wreak havoc. Even if a single false message goes undetected it may cause dire consequences and the number of vehicles that will receive a message would be considerable.

3.2.2 Services Related Applications: These are applications that aim to increase the comfort level or facilities for a traveler. These include automated payment services, internet availability, and multi-media services. Information services (like finding the closest fuel station etc). These applications are considered less important as of now as compared to the safety applications.

Both categories of applications require the communications to be secure, though the security requirements are of different nature. While the safety applications may only

require the authentication of the senders and integrity of data, applications like payment services require data privacy as well. In this work we shall consider security in case of safety applications only.

3.3 System Model Assumptions

We assume that vehicles will communicate with other vehicles and road-side units (RSU's). We also assume the existence of an authority and the vehicles can communicate to the authority through the RSU's.

Network Model: V2V (Vehicle to Vehicle) and V2I (Vehicle to infrastructure) communications over the wireless medium employ the *Dedicated Short range Communications* (DSRC) data link technology. Vehicles transit periodic messages on a common channel dedicated to emergency situations, among the available seven DSRC channels. As in DSRC, we assume that each vehicle periodically sends messages over a single hop every 300ms to all vehicles within a range of 10 seconds of travel from itself. These figures decrease in case of slowed down or stopped vehicles. Based on the content of the message a vehicle may decide to send a similar message on its own to other vehicles within its range. Since, every vehicle is broadcasting it is clear that all vehicles are supposed to receive messages very frequently and less frequently than it will send out messages.

Access to Road Side Units: A fixed infrastructure comprised of a number of base stations positioned in close proximity to highways will act as gateways to the internet and to some certifying authority.

On board communication unit: We assume that a vehicle has an on-board communication unit for V2V and V2I communications and are equipped with wireless technology based on IEEE 802.11 technology with which they can either communicate directly or use multi-hop communication.

Event data recorder: They provide tamper proof storage and will be responsible for recording the vehicle's critical data such as position, speed, time etc during emergency events. These data will help in accident reconstruction and the attribution of liability. These can be extended to record also the safety messages received during critical events.

Tamper proof device: It provides cryptographic processing capabilities. It will take care of storing all the cryptographic material and performing cryptographic operations, especially signing and verifying safety message .By binding a set of cryptographic keys to a given vehicle, TPD guarantees the accountability property as long as it remains inside the vehicle .The access to this device should be restricted to authorized people.

GPS: We expect that in near future, most vehicles will be equipped with GPS receiver providing fairly accurate geographical position coordinates. However, the existence of GPS like device is not mandatory for supporting security in VANET.

Message Formats: The messages are sent periodically and they include location and time and speed information corresponding to the information. Emergency messages may be sent in case of occurrence of an event.

3.4 Security Challenge

VANET represent fully distributed and self organizing networks of vehicle to vehicle and vehicle to roadside communication based on wireless communication. Moreover, VANET nodes are highly mobile which result in frequent change in network topology.

It is clear from the above enumeration of applications that security requirements for the various applications have significantly varying needs with respect to security. VANET can be vulnerable to attacks and jeopardize user's privacy, For example, an attacker could inject beacons with false information, collect vehicles messages, track

their location or infer sensitive user data. Moreover, the system should be able to establish the liability of drivers in case some life critical information is inserted or modified by an attacker but at the same time it should protect as far as possible the privacy of drivers and passengers. Therefore, in order to thwart such attacks security and privacy enhancing mechanisms are necessary, in fact, a prerequisite for deployment.

In this work we will only consider the security of above mentioned safety applications. It is a consensus that vehicles will broadcast messages from time to time to pass-on various kinds of information to other vehicles. As the messages will be broadcast and there will be no one-to-one communication between vehicles so privacy of the messages is not required, but the vehicles need to make sure that the information has been sent by an authentic node in the network. Following are a few attacks that can be employed by adversaries in VANET-:

False Information: The adversary can try and infuse false information in the network for personal gains or just to create havoc. The false information can be about one's own position or about the environment. This may also include replaying of an older valid message.

Masquerading: One node can pretend to be another node by using false identity to get away with false information attacks or for some other purpose.

Denial of Service: An attacker may try to jam the network by aggressively injecting spurious messages.

Tracking other vehicles: An attacker may try to track a particular vehicle with malicious intent based on the messages transmitted by that vehicle.

Following are a few properties the security protocol must possess in order to thwart the above mentioned and other attacks in VANET:

Authentication: Vehicles must be able to ensure that the message has been sent by a legitimate node.

Anonymity: While the authenticity of sender must be verified it is also imperative that the actual identity of the sender is not revealed. It must also be impossible to link two or more messages to the same sender.

Non-repudiation: No vehicles should be able to deny sending a message if it actually has. It is very important for fixing liabilities to the right vehicles.

Verification of Data: This may not directly concern the security aspect, but in cases like an attacker replaying an older valid message; it must be possible to discard such messages based on context and current information about the environment. Here, we are not concerning ourselves with the verification of content of the message.

3.5 State of the art

VANET (Vehicular Ad-hoc-Networks) is an emerging research area. Currently, most of the research in VANET is focused on the development of a suitable MAC layer with very few efforts focused towards security architecture and protocols for VANET.

The research on VANET security is just starting, with few pioneer papers so far. The most prominent industrial effort in this domain is carried out by Car 2 Car Communication Consortium [12], the IEEE 1609.2 working group [22], the NoW project [23] and the SeVeCom project [24] with all of them developing VANET Security architecture. Their common basic elements include the use of *Certification authorities (CAs)* and public key cryptography to protect vehicle to vehicle (V2V) and Vehicle to infrastructure (V2I) messages. It has now become an established consensus that Public Key cryptography is the way to go about for VANET. This is mainly due to the fact the messages are broadcast and one-to-one communication is not the norm. Due to this fact symmetric key cryptography will incur huge costs in frequent key establishment

procedures and they are also difficult to implement as the nodes are constantly on the move. Therefore here on we will concentrate on public key methods only. For all the perspective security protocols, message authentication, integrity and non-repudiation, as well as protection of private user information are identified as primary requirements.

On academic front, there are few publications describing the security architecture of VANETS, [13,14,15] but not may of them proposes specific protocols that considers all the practical requirements needed to secure VANET safety applications. Gerlach [17] describes the security concepts for vehicular networks. Hubaux et al. [16] take a different perspective of VANET security and focus on privacy and secure positioning issues.Parno and Perrig [20] discuss the challenges, adversary types and some attacks encountered in vehicular networks; they also describe several security mechanisms that can be useful in securing these networks. El Zarki et al. [19] describes an infrastructure for VANETs and briefly mentions some related security issues and possible solutions. The use of digital signatures in the vehicular environment is discussed in [18]**.**

Meanwhile, [26] mentions VANET as an application for group signature, that is, cryptographic primitives for anonymous authentication. This is a stronger property than pseudonymous authentication, as any two group signature generated by a node cannot be linked. A Group signature scheme is basically a method for allowing a member of a group to anonymously sign a message on behalf of the group. In [29] Bellare proposes a static group signature based on underlying digital signature and encryption scheme in which size of group parameters depend on the number of group members. It also provides theoretical foundations for the group signature scheme along with various security requirements that it should satisfy. Bellare in [30] proposes a dynamic group signature scheme which doesn't depend on the number of group members. Xuanwu [31] proposes another dynamic GS approach based on elliptic curve cryptography.

However, [21] proposes schemes for VANET security that relies on the concept of pseudonym authentication. [21] assumes the presence of certification authority which is vested with legal power to disclose node identities and is required to certify the keys of vehicular nodes. It proposes the following schemes for VANET security

a) *Baseline Pseudonym [21]:* Under this scheme every node is equipped with a set of pseudonyms (public private key pairs) along with public keys certified by a certifying authority. It uses a digital signature scheme like RSA for signing messages and attaches public key certificate for message validation.

b) *Group signature Scheme [26]:* In this each node is equipped with a group public key and its private signing key. Thus this scheme allows any node to sign messages on behalf of group without nodes identity being revealed to the signature verifier.

c) *Hybrid Scheme [21]:* The combination of pseudonym with group signature is basic element of this scheme. It uses digital signature for message authentication and group signature scheme for creating on the fly certificates of public key.

3.6 Motivation

As established in the previous sections, the security of safety applications in VANET require only authentication along with anonymity. There are virtually no anonymous authentication schemes that have been developed keeping the requirements and constraints of VANET in mind. Various frameworks have been proposed and all of them target the pre-available authentication algorithms that show an exemplary performance as far as security is concerned but are not so impressive when it comes down to the storage and computational complexity and ease of implementation. Thus it is reasonable to devote time and effort to the development of such a scheme. It is specially justified in the wake of the fact that VANET is something that is expected to be up and running within a decade from now.

3.7 Objective

In section 3.5 we outlined the various methodologies or protocols for ensuring security in VANET. Apart from the security a scheme provides, it is paramount to consider the costs it incurs in terms of time and memory usage. Also it is reasonable to assume that a vehicle would most probably be confined to one area most of the time, which leads us to another assumption that a group of vehicles (we consider the group to be large) will not be sporadically dynamic in its composition. Hence our target is to develop a secure group signature scheme that is more efficient than those currently available in the literature.

3.8 Conclusion

In this chapter we outlined an introduction to VANET, it applications, constraints and requirements in terms of security and otherwise. Then the problem taken up in this work and its motivations have been explained. In the coming chapters we propose a group signature scheme based for VANET along with its security analysis followed by analysis and comparison of its efficiency with other alternatives.

Chapter 4

GSCRT: A Group Signature Scheme

4.1 Introduction

Based on the background information and system model assumptions established in the previous chapter, we proceed to present a proposal for security in VANET. This group signature scheme is based on certain assumptions about the composition and dynamics of the network which we outline in the forthcoming section. The scheme is based on Chinese Remainder Theorem and it has been built-up upon the hierarchical access scheme proposed in [33] and then we proceed by explaining the theorem and then moving on to our proposal.

4.2 Preliminaries

4.2.1 Network and Infrastructure Assumptions: Apart from the assumptions stated in section 3.3 we assume that a region is divided into several groups of vehicles. This group is of course assumed to be much larger than the set of vehicles a vehicle can communicate with at some time. It means that at any time a vehicle will be able to send messages to and receive from, a set of vehicles that are also in the same group. We assume the existence of a group manager for every group, typically a government authority or some car manufacturer's agent. The role of the group manager will become clear in the forthcoming sections.

4.2.2 Chinese Remainder Theorem: The **Chinese remainder theorem** is a result about congruence's in number theory. Following is the statement of the theorem in one of its forms:

Suppose $n_1,n_2,\ldots n_k$ are integers which are pair-wise co-prime. Then, for any given integers $a_1,a_2,\ldots a_k$, there exists an integer x solving the system of simultaneous congruences -:

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

Furthermore, all solutions x to this system are congruent modulo the product $N = n_1n_2\ldots n_k$. i.e. the solution x is unique modulo the product $n_1n_2\ldots n_k$.

Following is an algorithm to find the solution given the relatively prime numbers $n_1,n_2,\ldots n_k$ and the set of residues $a_1,a_2,\ldots a_k$.

Let N be the product of the relatively prime numbers $n_1,n_2,\ldots n_k$. Let N_i denote the product of these numbers excluding n_i.

The number c_i is computed as follows:

$$c_i = (N_i)(N_i^{-1} \bmod n_i)$$

Then the solution x can be written as

$$x = (\sum (a_i)(c_i)) \bmod N$$

4.3 GSCRT

In this section we present a group communication scheme based on Chinese remainder theorem.

4.3.1 Proposal: Let there are k group members. There is one group manager per group who is in charge of generating the keys and distributing them to the members.

Public Information (known to all members and manager)

N_G - A prime number

Group Manager has following information:

N_o - Private (known to manager only) no. used to reveal identity of the message sender

Nd_i - Private (known only to manager) number required to distinguish the product used in construction of a particular $CRTK_i$ (will use a different Nd_i during construction of a particular $CRTK_i$, Manager need not store them).

Both Nd_i *and* N_o are prime numbers of order of 512 bits

Group Members:

Each member M_i is given the following information by the group manager.

N_i - A prime number known only to M_i

a_i - A random number ($< N_i$) known only to M_i used for sign verification

Pr_i = Product* Nd_i

Here Product = $\prod (N_j)$ which will be same for every user of the group and j € {0 ...k }.

$CRTK_i \bmod N_0 \equiv ID_i$

$CRTK_i \bmod N_1 \equiv a_1$

$CRTK_i \bmod N_2 \equiv a_2$

.....

$CRTK_i \bmod N_i \equiv a_i$

.....

$CRTK_i \bmod N_k = a_k$

$CRTK_i \bmod Nd_i = ad_i$ *(here* ad_i *can be any random number <* Nd_i*)*

$CRTK_i = <ID_1, a_1, a_2, a_3, a_4 \ldots\ldots, a_i \ldots a_k, ad_i>$ *(k+2 Tuple) as in CRT*

N_G - A number known to all members.

All N_i's, N_G and Nd_i's are prime numbers.

Note that this entity $CRTK_i$ is unique modulo Pr_i (follows from Chinese Remainder theorem).

All this information is available with a particular member.

4.3.2 Signature Generation:

To send the message the member creates a signature Y in the following manner.

$$Y \bmod Pr_i \equiv CRTK_i$$
$$Y \bmod N_G \equiv Hash(Message)$$
$$Y = < CRTK_i, Hash(Message) >$$

4.3.3 Signature Verification:

To verify the signature a member M_j does the following -:

$$X = Y \bmod N_j$$
$$If\ (X == a_j)\ the\ signature\ is\ verified.$$

It is important to note that the verifier does not need to and cannot extract $CRTK_i$ of the sender, to verify the authenticity of the sender.

4.3.4 Identity Extraction:

Only Manager will be able to reveal the identity of message sender by doing following operation:

$$ID_i = Y \bmod N_o$$

This ID_i then can be mapped to the actual identity of the sender.

4.3.5 Correctness:

In order to verify the receiver does the following check -:

If($Y \bmod N_i == a_i$)

We have to prove that in case of an authorized sender, this check does stand to be true.

$CRTK_i = (\sum a_j * ((Pr_i/N_j)* (((Pr_i/N_j)^{-1} \bmod N_j)))) \bmod Pr_i$ ---------- (1)

Where j varies from 0 to k+1 (assuming $Nd_i = N_{k+1}$)

$Y = (CRTK_i (N_G * (N_G^{-1} \bmod Pr_i))+Hash<Message>(Pr_i*(Pr_i^{-1} \bmod N_G))) \bmod Pr_i*N_G$ -(2)

Let Z be a number such that

$Z \bmod N_0 \equiv ID_i$

$Z \bmod N_1 \equiv a_1$

$Z \bmod N_2 \equiv a_2$

.....

.....

$Z \bmod N_k \equiv a_k$

$Z \bmod Nd_i = ad_i$

$Z \bmod N_G \equiv$ Hash<Message>

Let $N_{k+2} = N_G$, $a_0 = ID_i$, $a_{k+2} =$ Hash<Message>

Let $P = Pr_i * N_G$

Therefore Z can be written as-:

$Z = (\sum a_j * ((P/N_j)* (((P/N_j)^{-1} \bmod N_j)))) \bmod P$ where j varies from 0 to k+2

$Z = (\sum a_j * ((P/N_j)* (((P/N_j)^{-1} \bmod N_j)))) \bmod P + a_{k+2} * (Pr_i * (Pr_i^{-1} \bmod N_G)) \bmod P$,

j varies from 0 to k+1

Let Z = Z1 + Z2, where Z1 and Z2 are the two terms in the above equation

$Z1 = (\sum a_j * ((Pr_i * N_G/N_j) * (((Pr_i * N_G/N_j)^{-1} \bmod N_j)))) \bmod P$

Since Pr_i is a multiple of N_j,

$N_G^{-1} \bmod N_j = N_G^{-1} \bmod Pr_i$

$Z1 = (((\sum a_j * ((Pr_i/N_j) * (((Pr_i*/N_j)^{-1} \bmod N_j)))) * (N_G * (N_G^{-1} \bmod Pr_i)))) \bmod P$ – (3)

Now we have from equation 1

$\sum a_j * ((Pr_i/N_j) * (((Pr_i*/N_j)^{-1} \bmod N_j)) = qPr_i + CRTK_i$ for some integer q ----- (4)

From (4) and (5)

$Z1 = ((qPr_i + CRTK_i) * N_G * (N_G^{-1} \bmod Pr_i)) \bmod P$

$= (((qPr_i * N_G + CRTK_i * N_G) \bmod P * (N_G^{-1} \bmod Pr_i) \bmod P) \bmod P$

$= (((qP + CRTK_i * N_G) \bmod P * (N_G^{-1} \bmod Pr_i) \bmod P) \bmod P$

$= (((CRTK_i * N_G) \bmod P * (N_G^{-1} \bmod Pr_i) \bmod P) \bmod P$

$= ((CRTK_i * N_G * (N_G^{-1} \bmod Pr_i)) \bmod P$

$Z1 = ((CRTK_i * N_G * (N_G^{-1} \bmod Pr_i)) \bmod P$ --------- (5)

$Z = Z1 + a_{k+2} * (Pr_i * (Pr_i^{-1} \bmod N_G)) \bmod P$

$= ((CRTK_i * N_G * (N_G^{-1} \bmod Pr_i)) \bmod P + Hash<Message> * (Pr_i * (Pr_i^{-1} \bmod N_G)) \bmod P$

$Z=((CRTK_i * N_G * (N_G^{-1} \bmod Pr_i) + Hash<Message> * (Pr_i * (Pr_i^{-1} \bmod N_G))) \bmod Pr_i * N_G$

= Y from (2)

Therefore Y = Z

Hence,

$Y \bmod N_j = Z \bmod N_j = a_j$

4.3.6 Application to VANET: For deploying GSCRT in VANET, we assume the vehicles are divided in groups of 10000 each (this number may of course and accordingly the parameters will change). Therefore there are 10000 N_i's per group each of 80 bits, while N_0 and Nd_i are of 512 bits each. We assume that there are Road side units available at boundaries of regions so that when a vehicle travels outside its group it can contact the manager through the RSU and obtain new parameters for the new group.

4.3.7 Addition of a new Member: A new member will obtain its parameters directly from the manager. In this scheme the addition of member will require the inclusion of a new N_i-a_i pair, thus leading to a need to change the parameters of all other members. This implies that the scheme is truly static in nature.

4.3.8 Removal of a Member: The advantages of using a group signature scheme for VANET are accompanied by some challenging problems, notably certificate revocation. For example, the certificates of a detected attacker or malfunctioning device have to be revoked, i.e., it should not be able to use its keys or if it still does, vehicles verifying them should be made aware of their invalidity. In this particular proposed protocol, CRTK given to member vehicles can be considered as a certificate. Following is one of the approaches for such revocation:

Once the Trusted authority has decided to revoke certificate of a given vehicle M, it sends to it a revocation message encrypted with the vehicle's public key(assuming symmetric key communication)as in [11].After the message is received and decrypted by the TPD of the vehicle, the TPD erases all the keys and stops signing safety messages. Then it sends an ACK to the CA. All the communications between the CA and the vehicle take place in this case via road side units (RSUs). In fact, the CA has to

know the vehicle's location in order to select the RSU through which it will send the revocation message. If it does not know the exact location, it retrieves the most recent location of the vehicle from a location database and defines a paging area with base stations covering these locations. Then it multicasts the revocation message to all these base stations.

4.4 Security Analysis

In this section we analyze the security and robustness of our algorithm with respect to various requirements of a group signature scheme and a few attacks.

4.4.1 Anonymity: This property requires that a member should not be able to reveal the identity of another member from the signature that the later has sent. In GSCRT the identity is embedded into the key in the following way:

$$CRTK_i \bmod N_0 \equiv ID_i$$

The number N_0 is not available to any of the members (it is available only with the manager). N_0 is definitely a part of the products available with all the members. So to reveal the identity a member must be able to factorize any of the products or must guess N_0. The size of the product is around 100 kilo bytes with two factors of size 512 bits which makes it computationally infeasible to be factorized. Guessing N_0 correctly has a probability 2^{-512} as N_0 is a 512 bit number. This probability is certain negligible which leads us to the conclusion that GSCRT provides anonymity.

4.4.2 Non-frameability: This property requires that no member should be able create any valid signature that links to identity of some member other than his own. To create a valid signature that frames some other vehicle, a member needs to know all N_i-a_i pairs and N_0. Let us assume that from the product, the adversary is able to extract the smaller prime factors (Ni's). Then to frame another member, he must get N_0 and the identity of the

member. Getting N_0 is equivalent to factoring the product of N_0 and Nd_i, which is computationally infeasible. Therefore GSCRT provides non-frameability.

4.4.3 Unlinkability: This property requires that deciding whether two different valid signatures were computed by the same group member is computationally hard. In GSCRT the members receive Y which yields CRTK of the member and the hashed message as residues modulo Pr_i and N_G respectively. If the recipient is able to extract CRTK then he can definitely conclude that the messages are from the same sender just by comparing CRTK's. It must be noted though that even in this event it is not possible for him to determine the identity of the sender.

Now let us focus on the question that whether CRTK can be extracted from from Y. The following equality gives the relation between CRTK and Y:

$$Y \bmod Pr_i = CRTK_i$$

The parameter Pr_i, where i the sender, is unknown to all the recipients. If the recipient attempts at guessing Pr_i, we can simply strike it off as it is large number and therefore computationally infeasible to guess. This leads us to conclude that CRTK cannot be extracted from Y. Therefore under GSCRT different messages sent by the same member are unlinkable.

4.4.4 Traceability: This property requires that the group manager is always able to open a valid signature and identify the actual signer. In GSCRT a vehicle can create a valid signature that cannot be traced to any identity if somehow it extracts all N_i-a_i pairs. In fact extracting only N_i's will suffice as a_i's can be extracted using N_i's from any valid signature received from some other vehicle. N_i's can be extracted by factorizing the product but it seems infeasible even though N_i's are the small factors, as both the size of the product and the number of N_i's is large. But if the adversary forms a coalition or colludes with several

others members and they all share their information, it will definitely reduce the complexity of factoring the product. This complexity will decrease with increase in the number of colluding members. N_i's can also be determined as in attack mentioned later in section 4.4.5.2.

Therefore GSCRT does not provide *coalition-resistance* and does not provide traceability in all conditions. Note that the properties of anonymity, non-frameability and un-linkability still continue to hold even in case of colluding members.

4.4.5 Attacks

In this section GSCRT is analyzed with respect to some attacks. Note that many of the attacks mentioned in the first chapter are automatically ruled out due to above mentioned properties of GSCRT.

4.4.5.1 Insider Replay Attack: In this attack a member of the group intending to cause confusion by propagation of contextually incorrect information, replays a signed message that he has received from some other member. This attack only gains the stature of an "attack" if the message is replayed after a considerable amount of time. The attack can be easily thwarted by including the "timestamp" in the message. The recipient of a message can check using the timestamp whether the message is "too old" to be used.

Let us now look at the scenario when the attacker tries to modify the timestamp in the original message. This will lead the message signature to change as it includes the hashed message. To make the signature to comply with the changed message the attacker has to change Y accordingly which means he needs to extract CRTK of the sender and then recreate Y. This is not possible as the attacker does not know and cannot feasibly guess Pr_i of the sender.

4.4.5.2 Guessing N_i's: Each N_i is of 80 bits. As N_i's are primes, total number of possible values for it is definitely less than 2^{80}. Therefore it is not too difficult to guess N_i's by enumerating all possible values. Whether the guessed values are correct or not can be determined by storing a set of messages $Y_1 \ldots Y_k$ all from different members and checking the residues modulo the guess value of an N_i. If all yield the same residue, the guessed value is a correct one. It may be argued that determining whether two messages are from different members is not obvious, this problem may be overcome by storing a larger number of members or by using the location information in the messages to distinguish. For example if locations from two messages received during the approximately same time are far apart, it may be concluded that the senders are distinct.

4.5 Time Complexity

4.5.1 Basic Definitions: The following definitions of bit complexity for basic operations on integers have been outlined in [32].

Integer Addition

Bit complexity of computing x+ y for integers x and y is O (lg x + lg y).

Integer Multiplication

Bit complexity of multiplying integers x and y is O((lg x) (lg y))

Integer Division

Bit complexity of dividing integer x by integer y is O((lg x) (lg y))

Modular Integer Addition

Input: A positive integer N and integers x,y € Z_N = {0,...,N 1}.

Output: x + y (mod N).

The bit complexity of this problem is O(lgN).

Modular Integer Multiplication

Input: A positive integer N and integers x,y € Z_N

Output: xy (mod N).

Here the bit complexity is $O((\lg N)^2)$

Modular Inverse

Input: A positive integer N and an integer $x \in Z_N = \{a \in Z_N : \gcd(a,N) = 1\}$.

Output: $y \in Z_N$ such that $xy = 1 \pmod N$.

The bit complexity is $O((\lg N)^2)$

Modular Exponentiation

Input: A positive integer N, an integer $x \in \{0,...,N\ 1\}$, and any integer k.

Output: $x^k \pmod N$.

Using the method of repeated squaring, the bit complexity is $O((\lg k)\ (\lg N)^2)$.

As Trusted Authority or Group Manager has sufficient storage and computation power, so we are not taking in to consideration the time required to generate CRTK and other parameters needed to join the group. Our emphasis will be on estimating the time required to generate message signatures and verifying them.

4.5.2 Signature Generation Complexity:

We use Chinese remainder theorem while generating message signatures and so we need to look into the operations involved, in order to calculate the bit complexity of signature generation.

$$Y \bmod Pr_i = CRTK_i$$
$$Y \bmod N_G = Hash\ (Message)$$
$$Y = \langle CRTK_i, Hash(Message) \rangle$$

Now, we know that $CRTK_i$ and Pr_i are of order of (k'b) bits and size of N_i is b bits.

$$Y = (CRTK_i\ (N_G * (N_G^{-1} \bmod Pr_i)) + Hash\langle Message \rangle (Pr_i * (Pr_i^{-1} \bmod N_G))) \bmod Pr_i * N_G$$

Bit complexity Calculations:

*1. (N_G * (N_G^{-1} mod Pr_i)*

It involves Modular inverse of N_G w.r.t Pr_i which is of O $((\lg Pr_i)^2)$=**$O((k'b)^2)$** and multiplication of N_G with its inverse which is of O(k'b* b) = O($k'b^2$). However, size of this product is (k'+1)*b bits.

*2. ($CRTK_i$ (N_G * (N_G^{-1} mod Pr_i)))*

So bit complexity of ($CRTK_i$ (N_G* (N_G^{-1} mod Pr_i))) is O((lg $CRTK_i$) lg(N_G* (N_G^{-1} mod Pr_i))

=O(k'b * (k'+1)*b) = **$O((k'b)^2)$.** Size of this product is (2k'+1)*b bits

3. Pr_i(Pr_i^{-1} mod N_G))*

It involves Modular inverse of Pr_i w.r.t N_G which is of $O((\lg N_G)^2)$=$O(b^2)$ and multiplication of Pr_i with its inverse which is of O(k'b* b) = **$O(k'b^2)$**. However, size of this product is k'+1)*b bits

4. Hash<Message> (Pri(Pr_i^{-1} mod N_G))*

So bit complexity of Hash<Message>(Pr_i*(Pr_i^{-1} mod N_G))

is O((lg Hash<Message>) lg (Pr_i*(Pr_i^{-1} mod N_G))) =O(b*(k'+1)*b) = **$O(k'b^2)$**

Size of this product is (k'+2)*b bits

*5. ($CRTK_i$ (N_G * (N_G^{-1} mod Pr_i)) + Hash<Message>(Pr_i*(Pr_i^{-1} mod N_G)))*

Bit complexity of addition is O (Size of (1) + Size of (2)) = O (3(k'+1)*b) =O (k'b)

6. Message signature (Y)

Bit complexity of computing Y is equivalent to calculating of Modulus of (3) w.r.t Pr_i*N_G = O $((\lg Pr_i*N_G)^2)$ = **$O((k'b)^2)$**

Thus we can conclude that bit complexity of generating message signature is $O((k'b)^2)$

4.5.3 Signature Verification Complexity:

For each N_i and a_i

$X = Y \bmod N_i$

If ($X == a_i$) the signature is verified.

We only require taking modulus of *Y w.r.t. to* N_i and we know that modulus is similar to dividing *Y w.r.t. to* N_i and calculating the remainder.

Bit complexity of division will be of order O(lg (Y) lg (N_i)). We can approximate Y size to be order of k''b bits and we know size of N_i to be b bits. Hence bit complexity of verification is O(k''b * b)) = $O(k''b^2))$.

Note : Here k'' is equal to (k + c') and k' is equal to (k + c),

where k is number of N_i used while creating any CRTK and c, c' are constant.

4.6 Overhead and Storage Requirements

Let each N_i's have size b bits and there are k members. CRTK's and Pr's will have size of the order of (b*k+1024) bits. This poses a problem for large groups as CRTK and Pr will lead unacceptable size requirements. The overhead will be the size of Y which will of the order of size of Pr*N_G in the worst case.

For b = 80 bits and k = 10000 storage size is of the order of **202 kilo bytes** and overhead is of the order **101 kilo bytes.**

4.7 Conclusion

In this chapter GSCRT, a group signature has been proposed. The scheme does well on the security front, but does not provide coalition resistance. Though signature generation and verification involve fairly simple operations, the scheme still is not efficient due to the huge size of the parameters involved. We have analyzed for 10000 members as typically the number of vehicles in a group is expected to be large. In the next chapter we present a modified group signature scheme which performs much better comparatively.

Chapter 5

Modified GSCRT

5.1 Introduction

Because of the memory requirement problems with the original GSCRT scheme mentioned in the previous chapter, we propose a new version of the previous scheme which tries to overcome memory requirement problems. Following the specifications for modified GSCRT we show that this version performs drastically while providing the same security.

5.2 Modified GSCRT

In this scheme, the number of N_i used in the construction of the $CRTK_i$ is not equal to but less than the number of users (or vehicles) present in the group. Therefore the storage requirements will not increase linearly with the number of group members.

5.2.1 Proposal: Let there are n group members. Let k+2 be the number of prime numbers used to construct a particular CRTK

Public Information (known to all members and manager)

N_G - A prime number known to all members

Group Manager has following information:

N_o - Private (known only to manager) number used to reveal identity of the message sender

Nd_i - Private (known only to manager) number required to distinguish the product used in construction of a particular $CRTK_i$ (will use a different Nd_i during construction of a particular $CRTK_i$, Manager need not store them).

Both Nd_i and N_o are prime numbers of order of 512 bits

Group Members:

The manager creates k_C2 pairs of N_i's and corresponding a_i's , and distributes 2 $<N_i, a_i>$ pairs to each member.

Each member M_i is also given the following information by the group manager.

Pr_i = Product* Nd_i

Here Product = $\prod (N_j)$ which will be same for every user of the group and j € {0 ...k }.

N_G - A number known to all members.

All N_i's, N_G and Nd_i's are prime numbers.

$CRTK_i$ which is created as follows:

$$CRTK_i \bmod N_0 \equiv ID_i$$

$$CRTK_i \bmod N_1 \equiv a_1$$

$$CRTK_i \bmod N_2 \equiv a_2$$

.....

$$CRTK_i \bmod N_i \equiv a_i$$

.....

$$CRTK_i \bmod N_k = a_k$$

$$CRTK_i \bmod Nd_i = ad_i \quad (\text{here } ad_i \text{ can be any random number} < Nd_i)$$

$$CRTK_i = <ID_1, a_1, a_2, a_3, a_4 \ldots\ldots, a_i \ldots a_k, ad_i> \ (k+2 \text{ Tuple}) \text{ as in CRT}$$

The modulus is taken with respect to N_0 ,N_1..... N_i..... N_k, Nd_i.

Note: Nd_i is a prime number and thus can be used along with all other N_i in CRT.

All this information is available with a particular member.

5.2.2 Signature Generation: To send the message the member creates a signature Y in the following manner.

$Y \bmod Pr_i = CRTK_i$

$Y \bmod N_G = Hash\ (Message)$

$Y = < CRTK_i,\ Hash\ (Message) >$

5.2.3 Signature Verification: To verify the signature a member M_j does the following -:

For each N_i and a_i it has (there are two)

$X = Y \bmod N_i$

If ($X == a_i$) the signature is verified.

It is important to note that the verifier does not need to and cannot extract CRTK of the sender, to verify the authenticity of the sender.

5.2.4 Identity Extraction: Only Manager will be able to reveal the identity of message sender by doing following operation:

$ID_i = Y \bmod N_o$

This ID_i then can be mapped to the actual identity of the sender.

Note:

$N_0, N_1, N_2, \ldots\ldots N_k, N_G, Nd_i$ they are all prime numbers.

Here there can be multiple users using the same set of values for $<N_i, a_i>$ for verification but $CRTK_i$, they will use will be different because of different ID_i and Nd_i

Moreover, they will have different Pr_i because of Nd_i which will ensure that no other user can get back $CRTK_i$ from Y as he doesn't have any idea about Nd_i used and hence no knowledge of product (as modulus) used in Y.

5.2.5 Correctness: As underlying operations remain the same as mentioned in the original GSCRT, correctness proof is similar to the one mentioned before.

5.2.6 Application to VANET: For deploying GSCRT in VANET, we assume the vehicles are divided in groups of 10000 each (this number may of course and accordingly the parameters will change). The number of N_i's is taken to be 25 each of 80 bits, while N_0 and Nd_i are of 512 bits each.

5.2.7 Addition of a new member: In contrast to the original version the modified GSCRT scheme is not absolutely static in nature. When a new member has to be added, the manager can assign it to any of the existing k sets and will provide the parameters accordingly.

5.2.8 Removal of a member: Apart from the approach mentioned in section 4.3.8, there is one more approach based on timestamps for certificate revocation or removal of members.

***Other Approach*:** We can opt for using short certificate lifetime that will make certificates (CRTK's) expire thus revoking the certificates. This can be achieved by including a timestamp during the creation of CRTK by the trusted authority which specifies the date up to which a particular CRTK is valid.

Once the trusted authority has included a timestamp in CRTK that CRTK will remain valid till date. After CRTK expires, in order to continue communicating with that CRTK, vehicle has to go to road side units to get it CRTK refreshed with a new timestamp else vehicle will not be able to communicate with its expired CRTK. Thus, this approach can also be used to revoke a particular vehicle's certificate (CRTK) by not refreshing its certificate with a new timestamp. However, a malicious node will be able

to send erroneous message as long as its certificate is valid. Thus, creating a vulnerability window. However, this vulnerability window can be reduced by asking vehicles to frequently refresh their CRTK with new timestamps

5.3 Timestamp inclusion in $CRTK_i$

The manager creates k_C2 pairs of N_i's and corresponding a_i's, and distributes 2 $<N_i, a_i>$ pairs to each member. Each vehicular member M_i is also given the following information by the group manager.

Pr_i = Product* Nd_i. Here Product = $\prod (N_j)$ which will be same for every user of the group and j varies from 0 to k.

Manager includes a timestamp t_i by XORing it with all the a_i's in $CRTK_i$

Timestamp t_i is basically a date till which this $CRTK_i$ is valid. For this protocol its taken to be 10 days from the date on which vehicles comes to refresh its timestamp .So that the vehicles knows when its $CRTK_i$ will expire and can accordingly refresh its timestamp.

$CRTK_i$ which is created as follows:

$CRTK_i \bmod N_0 \equiv ID_i \oplus t_i$

$CRTK_i \bmod N_1 \equiv a_1 \oplus t_i$

$CRTK_i \bmod N_2 \equiv a_2 \oplus t_i$

.....

$CRTK_i \bmod N_i \equiv a_i \oplus t_i$

.....

$CRTK_i \bmod N_k = a_k \oplus t_i$

$CRTK_i \bmod Nd_i = ad_i \oplus t_i$ *(here* ad_i *can be any random number <* Nd_i*)*

$CRTK_i = <ID_1, a_1, a_2, a_3, a_4....., a_i... a_k, ad_i>$ *(k+2 Tuple) as in CRT*

The modulus is taken with respect to N_0 ,N_1,..... N_i..... N_k, Nd_i.

All N_i's, N_G and Nd_i's are prime numbers.

Note: *This $a_1 \ominus t_i$ is XORring of a_1 with t_i*
All this information is available with a particular member.

5.3.1 Signature Verification with Timestamp:

To verify the signature a member M_j does the following -:

For each N_i and a_i he has (there are two)

$X = Y \bmod N_i$

$Z = X \ominus a_i$

If (Z is a valid timestamp) the signature is verified.

5.4 Security Analysis

5.4.1 Properties: In the original GSCRT scheme we showed that GSCRT provides anonymity, non-frameability and unlinkability. These properties are unchanged as far as modified GSCRT is concerned. The scheme still does not provide *coalition-resistance* and traceability in all conditions. In fact achieving non-traceability becomes easier in the modified version because extracting small factors is easier as they are less in number for the same group size.

5.4.2 Attacks: Two attacks were mentioned in the section 4.4.5. Modified GSCRT like the original version is resistant against the insider replay attack. For the attack in section 2.4.5.2, the original scheme failed, but the modified GSCRT withstands that attack due to presence of timestamp in a_i's. Due to this when the adversary tries to guess N_i's, he is unable to verify correctness of the guess as even for the same N_i, different messages may give different residues due to possible presence of varying timestamps. The group

manager can in fact make sure that each member has a different timestamp. An attempt to try all possible combinations of N_i-a_i pairs will be computationally infeasible.

5.5 Time Complexity

5.5.1 Signature Generation: In this proposal, we do signature generation in a fashion similar to that in proposal 1. So, time taken to sign a message will be of order **$O((k'b)^2$**). But here k' is significantly less than that in proposal 1(reduction is from 10,000 to 25)

5.5.2 Signature Verification: Signature Verification is different from the one in proposal 1. In this scheme, vehicle has to verify signature for every $<N_i, a_i>$ it has. However, verification for different pairs can be done in parallel thereby keeping the verification time equivalent to verification by a single $<N_i, a_i>$ pair. Hence time taken for verification is of order of **$O(k''b^2))$** as given earlier.

5.6 Communication Overhead and Storage Requirements

5.6.1 Overhead: While sending the message, a vehicle needs to send Y also along with it in order to get verified and accepted. So, communication overhead consists of byte size of Y. Let the number of Ni's be k and each be of b bits. Let the number of members be n. Then CRTK's and Pr's will be of order (k*b + 1024) bits each. A set of N_i's and the corresponding a_i's will be shared by (n/kC2) members. We can choose k and b to reduce the storage and to restrict the number of members sharing the same set of N_i's and corresponding a_i's.

$$Y \bmod Pr_i = CRTK_i$$
$$Y \bmod N_G = Hash\ (Message)$$
$$Y = < CRTK_i, Hash(Message) >$$

For b = 80 bits, n = 10000 and k = 25

- Size of $CRTK_i$ is of order of 25*80 + 1024 = 3024 bits =378 bytes
- As N_G is of the order of 80 bites and Y is created using CRT we can say
- Overhead = Size of Y = order of 3024+80 bits = 3104 bits **= 388 bytes** in the worst case.

5.6.2 Storage Requirements: Storage size comes out to be = 3024 ($CRTK_i$) + 3024(Pr_i) + 320 bits (N_i's and corresponding a_i's) + 80 bits (N_G) = 6448 bits = **= 806 bytes.**
However, use of fixed numbers of Ni's while constructing $CRTK_i$ leads to sharing of same pair of $<N_i, a_i>$. Therefore, numbers of members with same parameters is approximately 33. i.e. (10,000/25_C2). Thus using the second scheme we can achieve considerable reduction in communication overhead and storage requirements.

5.6.3 Communication Overhead Comparison: In [21] for baseline pseudonym based approach, overhead consists of public key(25 bytes), certificate on public key (64 bytes) along with message signed with public key (48 bytes) , giving total overhead of 137 bytes. For Group Signature approach, overhead is signature on message signed with private signing key .Hence, total overhead is 225 bytes. Similarly for hybrid scheme overhead comprises of public key (25 bytes) , certificate on public key using group signature (225 bytes) along with message signed with public key (48 bytes) giving a total overhead of 296 bytes.

5.6.4 Storage Overhead Comparison: For baseline Pseudonym based approach mentioned in [21], one need to store certified public private key pairs. So assuming 8-10 hrs of daily car usage and refilling after one year , a vehicle needs to have approximately 200,000 such pairs where each pairs has size of 113 bytes (25 bytes public

key + 24 bytes private key + 64 bytes of certificate) , there by giving combined storage size of 22 MB. Security level for certificate is 128 bits.

For Group Signature Scheme, [21] uses a security level of 128 bits , therefore one need to store at least a group public key and a private signing key for signing messages on behalf of group. Therefore, total storage size

= 800 bytes (group public key) + 64 bytes (private key) = 864 bytes.

In hybrid approach, we have a digital signature on message with security level of 80 bits along with a certificate of public key created on the fly using a group signature. Therefore, we need to store at least a public and private key pair for digital signature scheme along with group signature parameters. Thus, total storage requirement is

25 bytes public key + 24 bytes private key + 800 bytes(group public key) + 64 (group signing private key) = 913 bytes.

Therefore our group communication scheme performs well on the storage front while it is marginally more costly as far as the communication overhead is concerned.

Scheme	Overhead (in bytes)	Storage (in bytes)
GSCRT	101 Kilo	202 Kilo
Modified GSCRT	388	806
Baseline Pseudonym (using ECDSA)[21]	137	22 MB
Group Signature [26]	225	864
Hybrid [21]	298	913

Table 5.1 Overhead and Storage Comparison with Other Schemes

5.6.5 Time Complexity Comparison

For quiet some time now, RSA is the most used and preferred public key encryption scheme for its security and simplicity. Many group signatures proposed till now are based on strong RSA assumption [27, 28]. There are others based on the Diffie-Hellman assumption or bilinear pairings [25, 26]. In this section we compare the time complexity of our algorithm with that of RSA. We assume 1024 bit for RSA and same before-mentioned specifications for our scheme.

The basic operation in RSA is modular exponentiation modulo the 1024 bit key. The generation and verification of signature employ similar operations hence we assume similar complexity for them.

RSA signature generation/verification complexity:

Generation: $Y = M^e \bmod N$ Verification: $M = Y^d \bmod N$

Complexity: $O(\lg(e)\lg(N)^2)$ Complexity: : $O(\lg(d)\lg(N)^2)$

	RSA	GSCRT
Signature Generation	$O(\lg(e)\lg(N)^2)$	$O((k'b)^2)$
Signature Verification	$O(\lg(d)\lg(N)^2)$	$O(k'b^2)$

Table 5.2 Time Complexity Comparison with RSA

The generation and verification complexity of RSA goes to $O((\lg(N))^3)$, if e and d are O(N). In comparison with our scheme k'b = O(lg(N)), leading to the fact that signature generation in GSCRT is comparable to that in RSA while it performs better than RSA when it comes to signature verification. As explained earlier lower verification time is suitable for VANET. Both provide the same security as in both cases the security comes down to factorizing an integer with two large prime factors (512 bits each).

5.7 Conclusion and Future Work

In this chapter we presented the modified GSCRT group signature scheme which performs much better than the original GSCRT scheme. The weakness of the algorithm towards coalition resistance still persists. Also the algorithm does not seem to be very scalable if the number of cars increases drastically. In that case the vehicles need to be redistributed into new groups which will require lot of effort. Redistribution will become unavoidable as with increase in the number of vehicles the size of the parameters involved will become too big to be acceptable. Another point to be considered is that all analyses here have been done keeping the worst case in mind. The size of Y, in reality and in general will be much less (even half) than the stated size.

The link between the theoretical security of an algorithm and the practical security required in VANET is uncertain to say the least. For example researchers around the world have said enough about the anticipated presence of Tamper Proof Devices in the vehicles to act as cryptographically secure storage for secret information. If this is truly the case then the underlying algorithm may actually stop worrying about coalition-resistance altogether as if the secret information cannot be extracted, they of course cannot be shared as well.

This algorithm presents something that is quiet against the norms prevalent in our times. Almost all public key algorithms coming are based on much more complex mathematical problems as compared to Chinese Remainder Theorem. We believe that to meet stringent efficiency requirements of VANET we will have to look beyond conventional methods and schemes. In its current form GSCRT is too raw to be accepted as a candidate to provide security in VANET, but it certainly does throw light on new areas and possibilities which are there to be explored.

Bibliography

[1]S. Nandi, B.K. Kar, and P. Pal Chaudhuri, "Theory and applications of cellular automata in cryptography", IEEE Trans. Computers, vol. 43, no. 12, pp. 1346- 1357, Dec. 1994.

[2] William Stallings; Cryptography and Network Security, Prentice Hall, 2003.

[3]D. Mukhopadhyay and D. RoyChowdhury, "Key Mixing in Block Ciphers through Addition modulo 2":' Cryptology ePrint Archive, 2005.

[4] E.Biham.New Types of Cryptanalytic Attacks using Related keys,Advances in Cryptology- EUROCRYPT'93,LNCS 765, Springer- Verlag, 1993,p 398-409

[5]P. Pal Chaudhuri, D.Roy Chowdhury. Sukumar Nandi, and Sanlanu Chattopadhyay, Additive Cellular Automata Theory and its Application,vol. 1, Chapter 2-4, IEEE Computer Society Press, 1997.

[6] L.Knudsen.Practically Secure Feistel Ciphers,Fast Software Encryption, First International Workshop Proceedings,LNCS 809,Springer-Verlag 1993,pp 211-221

[7]Joan Daemen and Vincent Rijmen ," The design of Rjindael , Springler –Verlag,2002

[8]Lauren May ,Matt Henricksen, William Millan "Strengthening the key schedule of AES "Proceedings of the 7th Australian Conference on Information Security and Privacy,2002

[9]A Novel Suite of Tests for Evaluating One Way Hash Functions for Electronic Commerce Applications Karras, D.A. Zorkadis, V. Euromicro Conference, 2000, Publication Date: 2000, pp 464-468 vol.2

[10]http://www.mathpages.com/home/kmath439/kmath439.htm

[11] M.Raya and J.-P.Hubaux, *Securing vehicular ad hoc networks*, in: Journal of Computer Security 15(2007), 39-682005, pp.11–21.

[12]http://www.car-2-car.org/.

[13] J.Blum and A.Eskandarian, *The threat of intelligent collisions*, IT Professional 6(1)(2004)

[14] Y.C.Hu and A.Perrig, *A survey of secure wireless ad hoc routing*, IEEE Security & Privacy 2(3)(2004),28–39.

[15] K.Sampigethaya, L.Huang, M.Li, R.Poovendran, K.Matsuura and K.Sezaki, *CARAVAN: Providing location privacy for VANET*, in: Proceedings of the Workshop on Embedded Security in Cars (escar)'05, 2005.

[16] J.-P.Hubaux,S.Capkun and J.Luo, *The security and privacy of smart vehicles*, IEEE Security and Privacy Magazine 2(3)(2004),49–55.

[17] M.Gerlach, VaneSe, *An approach to VANET security*, in: Proceedings of V2VCOM'05, 2005.

[18] L.Gollan and C.Meinel, *Digital signatures for automobiles*, in: Proceedings of Systemics, Cybernetics and Informatics (SCI)'02, 2002.]

[19] M.ElZarki, S.Mehrotra, G.Tsudik and N.Venkatasubramanianm, *Security issues in a future vehicular network*, in: Proceedings of European Wireless'02, 2002.

[20]B.Parno and A.Perrig, *Challenges in securing vehicular networks*, in: Proceedings of the Workshop on Hot Topics in Networks (HotNets-IV), 2005.

[21] G.Calandriello, P.Papadimitratos, J.-P Hubaux, A.Lioy, *Efficient and robust pseudonymous authentication in VANET*, in: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks, Canada (2007), 19-28

[22] IEEE 1609.2.IEEE trial-use standard for wireless access in vehicular environments-security services for applications and management messages,July2006.

[23] M.Gerlach, A.Festag, T.Leinmuller, G.Goldacker, and C.Harsch, *Security architecture for vehicular communications*. In WIT2005, Hamburg, Germany.

[24] P.Papadimitratos, L.Buttyan, J-P.Hubaux, F.Kargl, A.Kung and M.Raya, *Architecture for secure and private vehicular communications*. In ITST'07,Sophia Antipolis ,France

[25] D.Boneh, X.Boyen, and H.Shacham, *Short group signatures*, 2004.

[26] D.Boneh and H.Shacham, *Group signatures with verifier-local revocation*, in CCS'04, pages 168–177, New York, NY, USA, 2004, ACM Press.

[27] J. Camenisch, M. Michels, *A Group Signature Scheme Based on an RSA-Variant*, 1998

[28] N.Baric and B.Pfitzman, *Collision-free accumulators and fail-stop signature schemes without trees*, In W.Fumy,editor, Proceedings of Euro crypt 1997, volume 1233 of LNCS, pages 480–494, Springer-Verlag, May 1997.

[29] M.Bellare, D. Micciancio and B. Warinschi, *Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions*, Advances in Cryptology - Eurocrypt 2003 Proceedings, Lecture Notes in Computer Science Vol. 2656, E. Biham Ed, Springer-Verlag, 2003.

[30]M.Bellare, H.Shi and C.Zhang. Foundations of Group Signatures: *The Case of Dynamic Groups*, Topics in Cryptology - CT-RSA 2005 Proceedings, Lecture Notes in Computer Science Vol. 3376, A. Menezes ed, Springer-Verlag, 2005.

[31] X.Zhou, X.Yang, P.Wei and Y.Hu, Dynamic *Group Signature with Forward Security and Its Application*, in: Proceedings of the Sixth International Conference on Grid and Cooperative Computing (2007), 473-480

[32] Lecture 7, http://www.cs.uwaterloo.ca/~watrous/lecture-notes.html

[33] Xukai Zou , Byrav Ramamurthy , Spyros S. Magliveras, *Chinese Remainder Theorem Based Hierarchical Access Control for Secure Group Communication*, Proceedings of the Third International Conference on Information and Communications Security, p.381-385, November 13-16, 2001