Group Security of V2Vusing Cloud Computing Processing and 4G Wireless Services

BISWAJIT PANJA, DAVID MORRISON
University of Michigan-Flint Flint, MI
PRIYANKA MEHARIA
Eastern Michigan University, Ypsilanti, MI
BHARAT BHARGAVA
Purdue University, West Lafayette, IN
and
ATUL PRAKASH
University of Michigan, Ann Arbor, MI

Vehicle to Vehicle (V2V) interfaces will provide the future of safe driving. Implemented along V2I (Vehicle to Interface) setups, a driver will be more aware of their surroundings and traffic will be less of a hazard. Companies such as Google are trying to make cars that drive themselves through external sensors and internal maps. The security issues in V2V communication are vast and still being solved. Telecommunication companies have been working in turn to implement cloud computing to improve what they can offer their data users. They have been consistent with their improvements of coverage and data transfer speed. The creation of 4G LTE has shown great promise for what can be done with data transfers and minimizing tower loads.

This research paper takes a look at how the system can be protected against outside intrusions. It uses a Cloud to do computations, making it more difficult to steal data from the cars themselves. The system is designed to be in constant communication with the cloud to prevent any data theft, and presents methods for encrypting the data sent between the cloud and the cars. It also addresses the issue of tower load, by creating car groups to lower the difficulty of communicating in high traffic situations. These car groups communicate with one car, the lead car which is the only car that is communicating directly with the cloud. It sends and receives messages which are then distributed amongst the group, including but not limited to car locations, media and security keys. Cars that are separated from a group are treated as the lead car of a single car group, and cars that are completely separated from the system must first validate themselves through the cloud.

Keywords: V2V, Cloud Computing, 4G, Security

1. INTRODUCTION

Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) are a promising technology to improve safety and optimize traffic flow. Research has been done on identification of threats, specification of security mechanisms and how to adapt cryptographic primitives to a V2V/V2I model. There are a number of threats to the V2V/V2I system. A jammer can be used to stop any transmissions from traffic signals, because it remains in a local place, and can be placed relatively easily. Fake signals can then be broadcast, causing accidents if the vehicle doesn't know about hazards. Any node in the relay can also disrupt other nodes, causing the whole system to have problems. This also affects the driver's privacy, as their personal information could be accessed from the car. It is important to investigate vulnerabilities in the system. Some of the requirements of security in the system include authentication of transmissions, keeping data consistent, non-repudiation, and keeping clients private information secure.

Cloud computing has been revolutionary to the IT industry, and also used as V2C [Alexiou et al. 2013; Sagstetter et al. 2013; Alexiou et al. 2013; **?**], a Vehicle to cloud infrastructure that used the Network as a service (Naas) architecture. Cloud computing has been targeted to

large and medium scale operations and has a number of problems. The European project SAIL introduces a new solution, Naas to replace the three existing cloud types; Software, Infrastructure, and Platform as a service (Saas, Iaas, and Paas). There have been a number of related projects, including Intel's WiMax connected car and Toyota's LTE connected car. The government has allocated part of the spectrum [Crosby and Vafa 2013] for wireless vehicles. 4G has also been proposed to be used, but in order to do some a number of security issues must be addressed.

The key objective of inter-vehicular communication is to increase driver and vehicle safety [Tsuboi et al. 2014; Wang et al. 2014; Liao et al. 2013; Panja et al. 2014; Baldini et al. 2013]. There are both passive and active ways of improving vehicle safety. Passive ways include seatbelts and airbags, while active ways include steer assist and lane reading. Inter Vehicle Area Networks (VAN) can share such information between vehicles. This includes Vehicle to Vehicle (V2V), Vehicle to Broadband Cloud (V2B), and Vehicle to Infrastructure (V2I). V2I communicates to roadside units (RSU), sending and receiving information.

Cyber security on non-computers is a new and growing issue. Modern cars have more intelligent MCU(micro controller unit)'s, more code, and therefore more risks to cyber-attack. Initially, companies made wireless communication devices in cars for simple tasks, such as emergency contact and gps. As we continue to push forward, they want to include V2V(vehicle to vehicle) communication and vehicle to device(cell phone, mp3, etc.) communication. From July to November 2011, malware on Android increased 472%. Once they communicate with a vehicle, they could easily infect it. The CVSS (Common Vulnerability Score System) can be used to check how vulnerable V2V system is. 376,000 vehicles of a certain model were built in the US and Canada in 2009.

The purpose of this paper is to implement a useable security solution for a V2V (Vehicle to Vehicle) network. We propose an approach for group security.

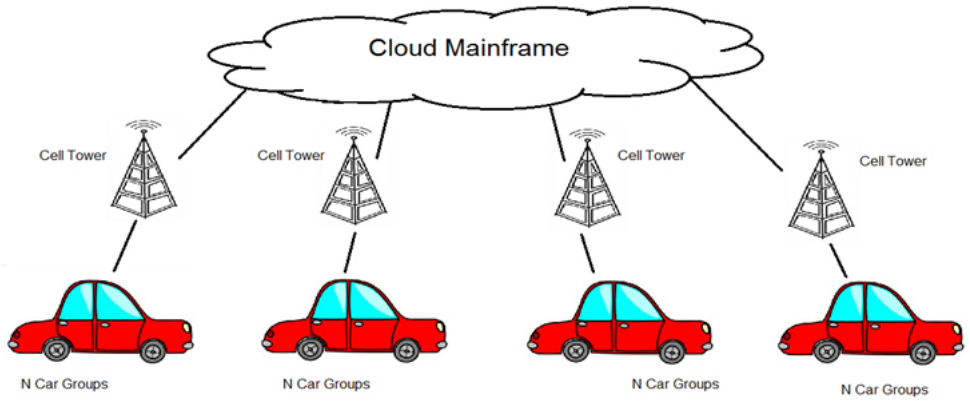


Figure 1

A cloud is responsible for storing all information and doing most of the calculation, seeing as it has more processing power. Each cell tower is connected to the cloud, and relay information to and from it. Each cell tower connects to a certain amount of "Car Groups". Each car in the group is connected to each other car, and the group is connected to the tower through the car with the best connection. This keeps every car connected, as long as it is close to a car with a connection, in cases where one loses connection in a tunnel. This requires a secure check to link into a group. If the car isn't in the group, it will be in its own group and connected to the tower. This means that joining a group can be done from the cloud. Every few minutes, each car will receive a security key. This key will be stored with the vehicle identification number. If the car is not in a group and loses its connection, the car may rejoin if its key matches that stored in its VIN, because it will be unable to change its key without a connection. The key will have to be

stored in the car in such a way that it is impossible to extract it, or you could easily connect any vehicle by sending a fake VIN and key. We would use a container that incinerates its contents if it is opened. Protecting the connection to the cloud is extremely important in this system. If for any reason it was tampered with, it could send harmful programs to the cloud.

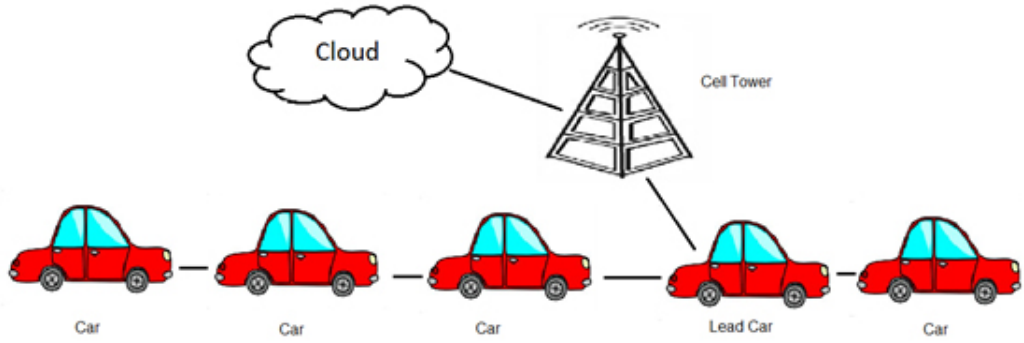


Figure 2

This paper is split up into 4 sections. First, some background to the problem will be introduced. Second, we will lay out a V2V network in which our security scheme is planned around. Next, we will introduce our security theme. Finally, we will look at any potential problems with our scheme.

2. RELATED WORK

SEVECOM [Leinmuüller et al. 2006] is the first phase of a long term undertaking of implementing V2V. SEVECOM investigates how to detect intrusions, how to keep data consistent and how to secure your position at all times. It looks into designing a user interface and cryptographic primitives that address issues in inter-vehicle communication. Implementing security services requires the ability to store sensitive data. However, service providers and owners should not have access to some of this data. The solution is creating tamper resistant modules. These might include environment detection sensors (detect abnormal changes in environment around it), which are a Level 4 security device. A level one device might just be a physical protection, like a difficult to open casing. Smart cards are considered to be trustable and provide decent protection, but may not provide enough protection. On the other end of the spectrum, the IBM 4758 PCI board provides a lot of protection, but is also extremely expensive. It will take time to find a balance between the right level of security and the price of the system. It is important to choose a Public Key Cryptosystem (PKCS) with decent implementation overhead and a decent execution speed. Also one that has the proper key, signature, and certification sizes. When choosing the PKCS, the Elliptic Curve Digital Signature Algorithm (ECDSA) and the NTRUSign outperform the RSA sign.

Cloud based navigation [Rangarajan et al. 2012] allows requests from the user to find routes with low traffic, or the fastest way home. This can be managed better by infrastructure providers, who can manage large data centers to compute multiple routes, specified by the automobile user. A major advantage of the new network capabilities is the ability to send street views of the area. Infotainment includes being able to have video conferencing from the car, as well as multimedia streaming. Compared to the three major models (Google, IBM and Eucalyptus), the V2C model fairs as well or better in Access Control, Packet and circuit switched network convergence, On-Demand network provisioning, Packet-based access quality, Auditing and assurance function, Stateful VM migration and Software network abstraction place. It does however lack Multi-Level Security, as it is in its early stages still. This paper proposed a V2C model as a solution for Vehicular communication.

While Vehicle to Vehicle (V2V) communication [Mangel and Hartenstein 2011] could be done with dedicated short range communication, it could also be done with cell networks to possibly provide better coverage. This paper attempts to predict how much load would be on each tower cell. Using UMTS (3G) or LTE (4G) communications to provide coverage for Cooperative Awareness Messages (CAM), cars will be able to communicate location, speed, and heading. UMTS was found to be barely able to handle 1500 CAM transmissions per second in a single channel, but LTE should be able to provide sufficient coverage. The capacity for a cell system to support this system depends on a number of factors. Having an available bandwidth, being able to broadcast and receive separately as opposed to together, being able to split load into multiple channels, and how many vehicles are in each tower range. Currently, no load problems have arisen from general traffic information and hazard warnings on LTE and UMTS. However, load demands go up as cars communicate. To get data, Munich's cell tower locations were found, and matched to the providers. GSM and UMTS are what are currently used, and while GSM is too old to use, it is on a frequency similar to LTE so it is analyzed. A Voronoi diagram composition splits each area into cells, which are covered by a certain number of towers. These will be used to check load on towers. Cells are thrown out if they are too large or too small for use.

Almost all research citeCalandriello2011 on Vehicular Communication (VC) has been focused on beaconing a status, and even so this can leave VC systems vulnerable. Projects such as Sevecom, Now, and Car to Car communication consortium have been working on security solutions. Security mechanisms protect traffic signals sent by each vehicle around every 1-3 hundred ms. However, under dense network conditions, security becomes difficult, posing the question are secure VC systems practical? This paper aims to answer that. There are 5 things that should be considered. Communication technology, system resources, network configuration and environmental factors, security protocols and supported applications. A baseline pseudonym (BP) scheme is where the Certification Authority (CA) issues short-lived keys that don't identify the vehicle, then sign messages with the corresponding key. Group Signatures (GS) are proposed as each vehicle self generates its own keys. However, they wish to use Hybrid Pseudonyms (HP). HP is when a node (car) generates its own keys, and each key is validated as a group, so that the vehicles are not identified. There are a few optimizations that can be made for both BP and HP. All tests were performed with Centrino 1.5 GHz. While this is powerful compared to what current systems use, it is a test for future systems. Communication reliability is very important to the solution, and depends on the channel properties and load. The heavier the channel load, the more likely for collisions to occur.

Controlling multiple Unmanned Autonomous Vehicles(UAV) using Spatially Secure Group Communication(SSGC) is the framework of this paper. Unmanned aerial vehicles have been deployed for many missions, and typically collaborate to exchange time critical information. However, a proper communication form is vital to controlling them as a group. This communication brings new problems. Communication congestion increases as the group size increases. Long range communication such as satellite communication is easier to hack into, so it is not preferable to close range communication. This paper defines the SSGC problem, presents an analytical model, investigates secure UAV communication and proposes a distributed solution. Relations between UAV systems and animals have inspired a solution based on nature.

Cloud computing is the newest way for businesses to share information. It is easier to use and control then older methods. With a new method brings new challenges. In order to protect the client, the Service Level Agreement (SLA) is the first line of defense.

The telecommunication industry [Zhu and Martinez 2012; Leinmuüller et al. 2006; Seong-Woo and Seung-Woo 2012] has helped turn the internet into a mobile environment and must embrace the cloud in order to remain competitive. Outlined in this paper are the challenges that must be addressed in order for this to take place. While telecommunication companies could forward data from clients to cloud services, a unique opportunity arises for the telecommunication companies to offer cloud services directly to clients. The data retrieved from working with their clients

will give them an advantage over other cloud based services. In order to use cloud services, the telecommunications industry will have them connect to Evolved NodeBs(eNBs) in the case of 4G LTE, which can be shared among multiple Mobile Network Operators. The eNBs is what sends out the signal for 4G LTE, so it is important that the cloud used can isolate data from different users and routed due to pre-determined rules. There are a number of challenges that come with cloud computing in the telecommunication industry. The first is a mix of legal and regulatory challenges.

3. PROPOSED APPROACH

This system has a number of difficulties it must overcome in order to be put into place. First, all cars must be able to communicate with a single cloud. This means that car companies would have to agree on using one cloud, or that the cloud would have to be implemented by the government for road safety. If car companies did use separate cloud frameworks, they would have to communicate with each other in order to have a complete road safety map. Second, this system requires a decent amount of cell towers at a level of 3g communication or higher. This means that areas that are very rural will have more trouble connecting. This is actually not bad, because most rural areas have fewer people and therefore less vehicles to monitor, lowering tower load. Third, a car which loses its ability to store data i.e: hard drive crash would have to be added to the network through a car dealer, because it would no longer have its key to rejoin the cloud. While dealer repairs are expensive, most car shops do not have the tools to fix many computer problems in cars, so this would not be too big of an issue. One final problem is propagation delay. The cloud will be solving problems that occurred nanoseconds earlier and then sending back replies a number of nanoseconds later. If the car is gathering the data itself, data coming in will be as late as a second before being gathered, sent, processed, and received. At 60 miles per hour your car will travel 88 feet before it receives this data. This problem can be solved by using better data transmission technology such as 4g LTE and minimizing tower load. It can also be helped by improving technology on data retrieval.

Notations used: i = *information packet*
K = *key private*
$K1, K2$ = *key public*
$E_{K1}(i|RSA)$ = *encryption function private*
$E_K(i|Cmac)$ = *encryption function public*
$D_K(i)$ = *decryption function*
L = *lead car*
L_S = *lead solo car*
C_n = *car group n*
c_{nm} = *car n in group m*
T = *cell tower*
S = *cloud server*

The system this research examined was a vehicle to cloud interface. The cloud creates groups out of each set of cars in a tower radius. S creates $C_1, C_2, \ldots C_n$ with cars $c_{1n}, c_{2n}, \ldots c_{mn}$. The size of the group will be based on the current load of the tower, with larger groups as the load goes up. In each group, the cars communicate with each other. The car with the best connection within a certain tolerance will be considered the lead car ***(L)*** and therefore the only one that communicates directly with the cloud ***(S)***. This allows cars in tunnels or with poor reception to communicate through other members of its group.

This design is as much for use as security, because as long as the car is connected to the cloud, it is much harder to tamper with it. The cloud will communicate using IEEE802.11p [2, 3] the current protocol for this type of system. If a car is alone or disconnected, it is treated as its own group, and the cloud searches for a new group to add it to based on gps information. The tower is connected to a Platform as a Service (PaaS) cloud structure which uses its software to

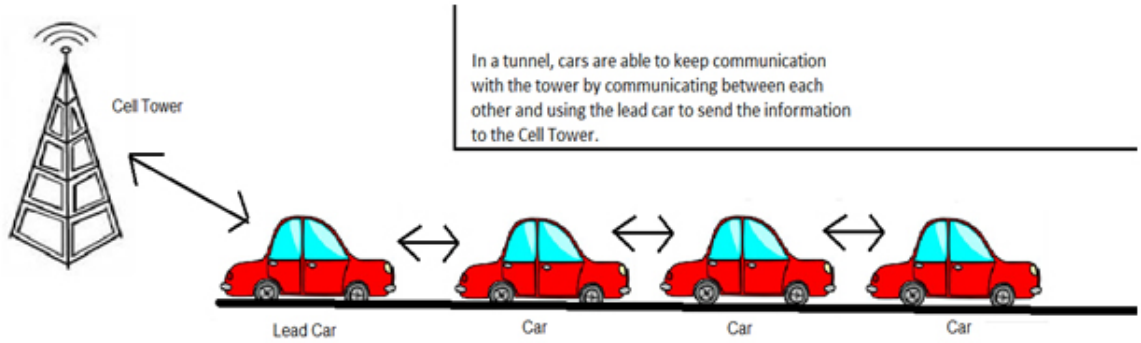


Figure 3

make all decisions the cars need. The cloud is responsible for making groups out of the cars, monitoring car positions to prevent collisions, streaming media applications and even generating security information such as keys.

Information that will be sent from a car to the tower will be in the form of a set of GPS coordinates. These will be in a form similar to ($xxx\ x_2x_2\ x_3x_3$, $yyy\ y_2y_2\ y_3y_3$) where (x, y) are our longitude and latitude coordinates respectively with (x_2, y_2) representing minutes and (x_3, y_3) representing seconds. The information will be used to determine where a car is at a given time, allowing the cloud to perform any calculations it needs. The information returned will be a list of cars in the group, as determined by the cars ip addresses. It would also include information about the lead car. This would be sent in the form $(z_1, z_2, \ldots, z_n)$, for a group of n cars where z_1 is the lead car. These data sets will be sent as packet iin our system. If a car needs to be added to the group, a set of keys will be sent out in the form $(z_{newcar}, D_K(i))$ which ensures that the incoming car can be recognized, and its message can be decoded. Upon entry, the cloud will send a new set of keys to the lead car for distribution. Data is frequently moving in a V2V system. Initially, data will be sent from the cloud to the tower. S uses $E_K(i|CCM)$ $\rightarrow T$ which uses $D_K(i)$ to decrypt. This will be passed using a land line as cipher text. It will be encrypted using CCM, with a new key generated at a regular interval. From here, the tower will send the information to the lead car, using a public encryption. This means that the tower will require a computer of its own, in which to convert. The keys for the tower and lead car will be cycled regularly, based on the towers load information and the group size in question. The data will be encrypted using RSA , with cMac authentication, and the Ctr encryption mode. T uses $E_{K1}(i|RSA)$ and sends it to L, which decodes it as $D_{K2}(i)$. Each group will communicate using a private key. L of group n will send out its messages encrypted $E_K(i|CCM)$ and sent to each car to decode $D_K(i)$. This key will be sent from the tower and cycled whenever the group status changes, for example when a car is added or leaves, or when the lead car is switched. The key will then be distributed amongst the group by the lead car. If a car is alone, it is treated as a lead car L_S and sent a set of public keys. The cloud will also recognize it as a solo car so it will not receive a private key. S sends $E_{k1}(i|RSA)$ to L_S and L_S uses $D_{K2}(i)$.

In order for this car to join a group, the cloud must determine that the car should be added. At this time, the lead car of that group and the solo car will each be sent a private key.

From here handshake protocol can take place between the two cars. LSuses EK(i — CCM) and sends it to L to decrypt using DK(i). The solo car will not be able to communicate using other cars in the group, so it must be close enough to the lead car.

After handshake protocol, the lead car or new lead car (if the solo car had a better connection) will be sent a new private key for the group and distribute it out. If the solo car becomes the lead car it will send the key to the old lead car to distribute.

If a car is disconnected from its group and the tower, it will store its latest key (public or

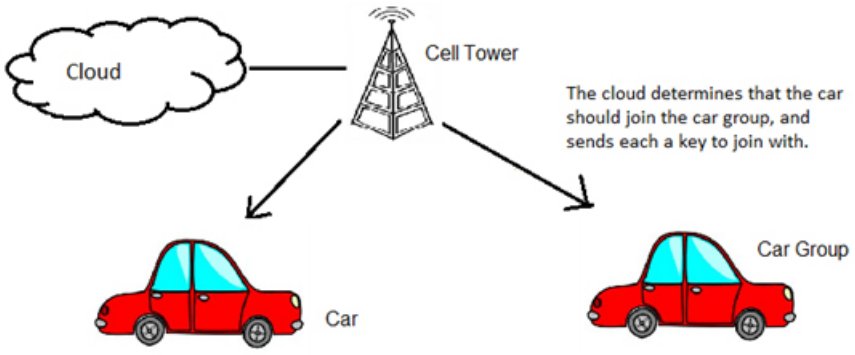


Figure 4

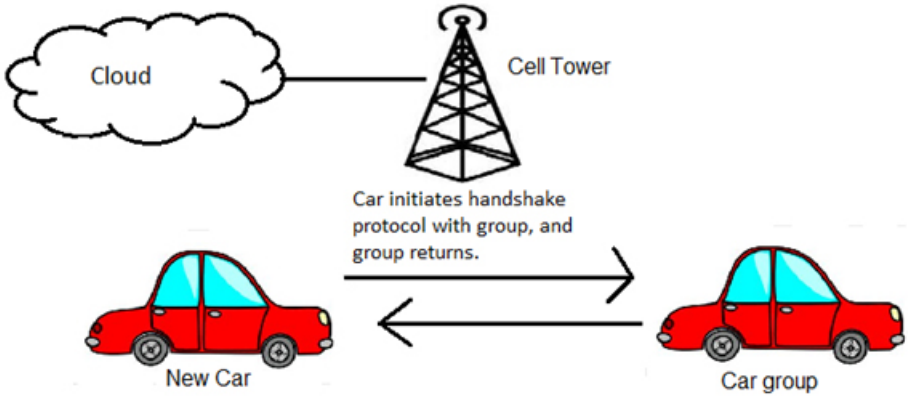


Figure 5

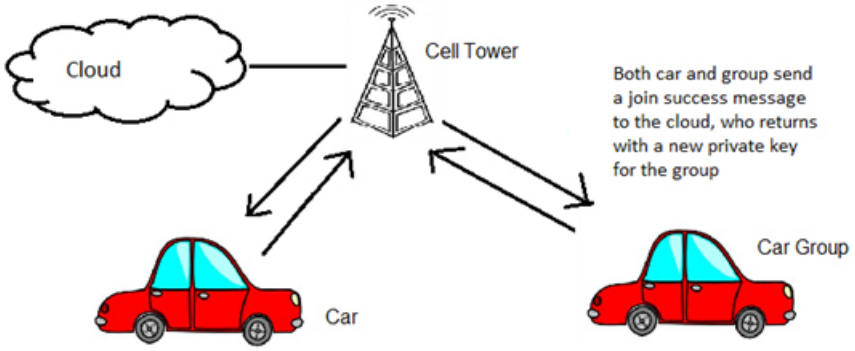


Figure 6

private) and continue to search or a signal. Once a signal is found, the car will send out a join request to the cloud in plain text and an encrypted handshake protocol. This will consist of the vehicles VIN number. The cloud will look under the vehicles VIN to find the last key the car was using and decode the handshake. It will then return its own handshake and a public key for the car to begin using. From here, it will be a solo car until it joins a group. By cycling keys

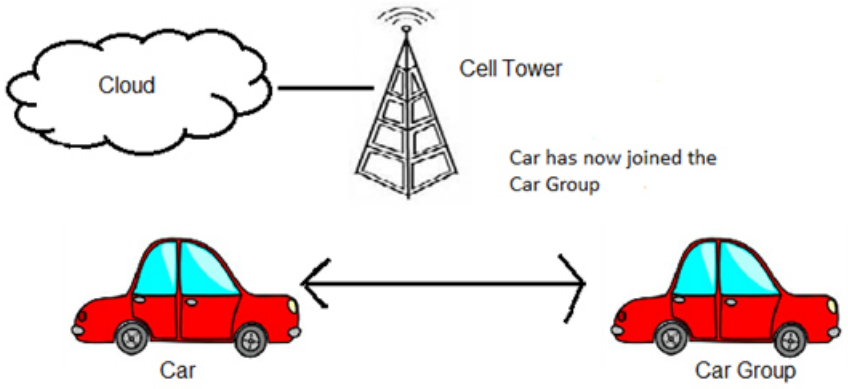


Figure 7

every time a car enters or leaves the group, the goal is to make every separated car have its own unique key. RSA was chosen due to its wide usage making it easier for all cars to use the same system if multiple companies are using the same network.

4. ANALYSIS AND MESSAGE FLOW

There are a number of functions that will need to be performed by both the cloud and the group of cars in order for this proposition to be feasible. The cloud needs to be able to calculate the location of each car, and track how groups will be formed. In order to do this, let us define a set of cars $C = (C_1, C_2, \ldots, C_n)$ in some group being tracked by the cloud, with car C1 being the lead car. These cars will send information in the form $G = (G_1, G_2, \ldots, G_n)$, where G is a set of GPS coordinates. The cloud will therefore receive set G for its associated set G. The cloud will use its new information G, along with the previous auto locations G' to create a set of vectors V. Given G_i and G_i', we can produce V_i in V. V_i = ¡$(xxx - x'x'x', x_2x_2 - x'_2x'_2, x_3x_3 - x'_3x'_3)$, $(yyy - y'y'y', y_2y_2 - y'_2y'_2, y_3y_3 - y'_3y'_3)$¿ where x and y are latitude and longitude respectively. Vector V_i will therefore be in the form ¡$(X, X_1, X_2), (Y, Y_1, Y_2)$¿. Another function the cloud will need to perform will be the creation of groups. This will be done based upon the tower load at the time. We will start with a variable D in the range of 5 to 15 as an indicator of the change in tower load.

From here, we will use some variable S as a 0 - 10 ranking to determine the current load of the tower. The cloud will compute the ceiling of $(D \times S)$ / 10, using that value to determine whether to increase or decrease group size. If ceil[$(D \times S)$ / 10] ¡ Cars in groups remove a car from each group. If ceil[$(D \times S)$ / 10] ¿ Cars in groups add a car to each group. Otherwise leave the groups alone. This will ensure that the tower load is not too high. Groups will be formed using a greedy function. Each cars connection will be looked at as a set $C_{conn} = (C_1, C_2, \ldots, C_m)$ for some m cars connected to the tower. The connection of each car will be ranked, with the best connections at the top of the list. From here, the top O cars, where O = the number of cars per group / m will be selected for use as lead cars. Once these cars are established as leads, the distance of each car from a lead can be established. For some car with coordinates $C = (xxx\ x_2x_2\ x_3x_3,$ $yyy\ y_2y_2\ y_3y_3)$, we can find its approximate distance from a lead car C' by $((xxx - x'x'x')2 +$ $(yyy - y'y'y')2)1/2$, $((x_2x_2 - x'_2x'_2)2 + (y_2y_2 - y'_2y'_2)2)1/2$, $((x_3x_3 - x'_3x'_3)2 + (y_3y_3 - y'_3y'_3)2)1/2$. Using this method, we can determine the cars nearest to each lead car. Once we have this, we can form groups equal to the ceil[$(D \times S)$ / 10]. Our private encryption will be RSA. Our key will be generated by as a set of prime numbers U_1, U_2. This will be our private key. These numbers will be generated in the range 10^{100} to 10^{200}. We will use these to generate our the first part of our public key $P = U_1 \times U_2$. P is the modulus for our keys, and its length will be the key length. From here, we will compute some n such that $n = (U_1 - 1)(U_2 - 1)$. We will use this number to

generate some e such that $e < n$ and $gcd(e, n) = 1$. This will be sent as part of our public key. To encrypt our message M, we will compute $C = M^e(modP)$. C will be our encrypted message. C will then be returned, along with our public and private key. It is now

time to decrypt this message. We create some $d = (1mod(n)/e)$. We then perform a binary expansion on our d. This will give us some set of powers of 2, $(d_1, \ldots, d_t)$ where t is the last term in the expansion. For each d_t, we will calculate $C^{d_t}(modP)$. This will create some values $(P_1, \ldots, P_t)$. Finally, we will multiply $P_t \times Pt \times \cdots \times P_1(modP) = M$. This will be our decrypted message. For CMAC encryption, we will begin by creating a message of length W_c, where W is the Cipher Block Length and c is constant. This message will be split into W bits. We then start with bit one and generate code $c_1 = E(W, W_1)$ where our bits are $W_1 \ldots W_w$. We then recursively work through the $c's$ saying $cn = E(W(W_1 \oplus cn_{-1})))$. Our code $T =$ the s leftmost bits of the bit string of bit length T. Our public key will be a set of two. The first part will consist of $K_1 = Lx$ and $K_2 = L \bullet x_2 = (L \bullet x) \bullet x$ where multiplication $(\bullet)$ is done in the finite field $(2n)$ and x and $x2$ are first and second order polynomials that are elements of $GF(2n)$.

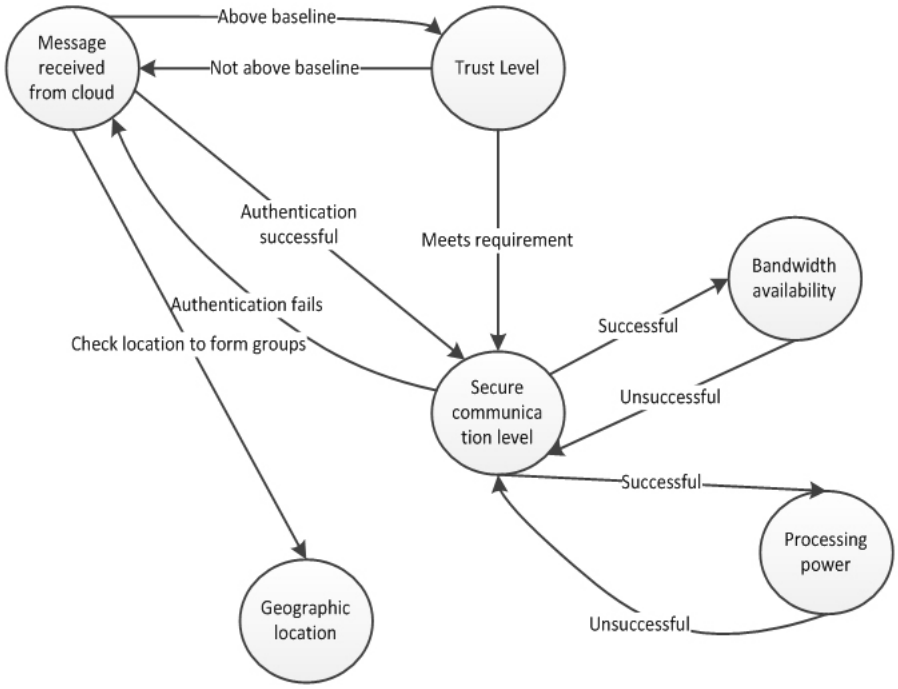


Figure 8

The state diagram shows the steps for checking 1. Trust level 2. Bandwidth availability 3. Processing power, in order to form groups. The starting state in the model is the "message received from cloud". The bandwidth, trust level and process power availability must meet the requirement provided by the cloud. We have not showed any error and accepting states, because we assume that each transaction for forming groups is atomic, meaning the parameters discussed must meet the requirement or the transaction has to start over.

The message flow diagram is similar to state diagram with the steps for formation of groups. Again, the requirements given by the cloud must be met in order to form groups. One of the blocks which may have not discussed enough is the "location information". This block helps to find group of vehicles based on geographic information. All the groups are formed using the

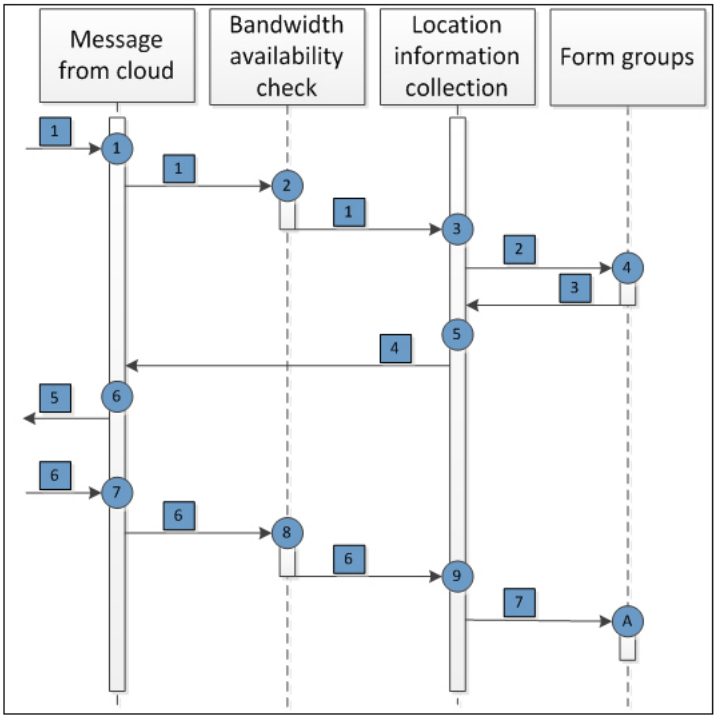


Figure 9

location behavior of vehicles. For example, if certain vehicles tend to stay in Ann Arbor, MI for 5 out 7 days, they may belong to a group rather than mixing vehicles from other region say Detroit.

5. IMPLEMENTATION AND EXPERIMENTAL ANALYSIS

This section provides the implementation details and experimental analysis of the proposed approach. We start with a class diagram, which shows the major classes, data members, member functions and interaction among themselves.The main classes we identified as 1. Cloud 2.Tower 3.Parent car 4.Child car. The private and public member functions are shown in each of the classes.

The purpose of theexperiments is to test a cloud's computing ability for creating groups on a single tower, to see how much computing power will be required for a network of multiple towers.We start with an arbitrary cloud, set up with our software for creating groups and managing our V2V system. We set up a map for the tower area which the cloud use to operate. The cloud then sendsconstantly updating coordinates from make believe cars. First, the cloud plot them on the map, and second it begins to group them. We set an upper limit for the tower load and the cloud make groups according to that size limitation. As we continue, we add more cars to the map which requires the cloud to continue adding to the groups. As the cloud computes, it sends keys to its theoretical cars, which received by a computer we use to monitor its response times.We observe the speed in which the cloud is able to return keys for its created groups, and figure out the cloud size required for a nationwide system.

In order to analyze if there are significant differences in the amount of packets dropped based on the number of cars in a system, we began by setting up an experiment in NS-2 to cause dropped packets. We set up a wired connection between a number of nodes, which varied for each trial of the test. We then passed packets over this connection and observed how packets

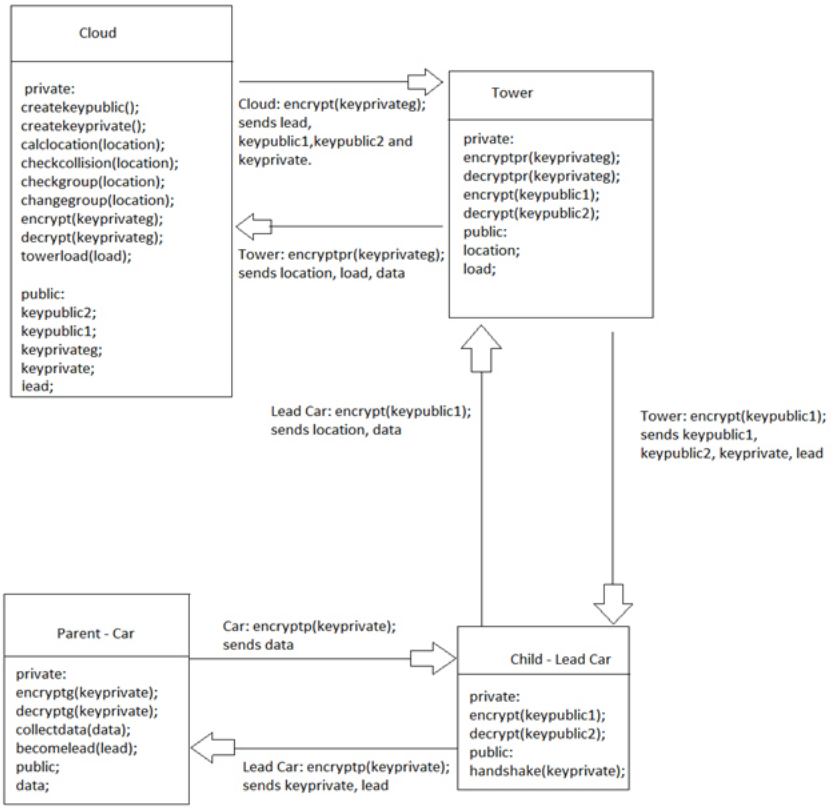


Figure 10

were dropped.

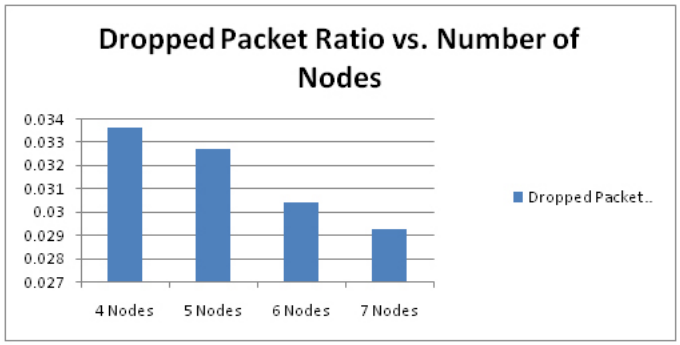


Figure 11

The above figure shows that the ratio of dropped packets to passed packets actually reduced as the number of nodes increased. The data presented here shows that as nodes were added, the ratio of completed to dropped packets is reduced. These tests were run using Droptail to determine packet dropping. Packets are size of 100000 bits. The experiment used CBR over UDP and FTP over TCP for two nodes communicating into one. Initial nodes used 10ms lag while all others use 20 ms of lag. The ratio of dropped packets varies by about .006, or .6% of our passed packets. This shows that while there is a little variation in the amount of dropped packets, it is not a significant amount. We see no significant amount of packets lost with each change of

size. However, for each set we notice that there are some dropped packets. This leads to a new question. What is the minimum bandwidth required for no packet loss in each of our group sizes.

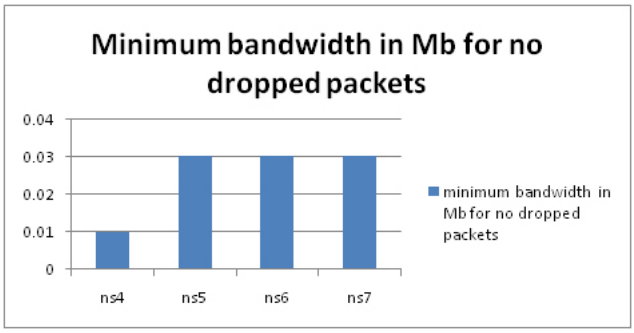


Figure 12

Above is our completion of a test to show what the minimum bandwidth is to prevent packet dropping. We see that the 4 node test requires a significantly lower amount of bandwidth to the 5, 6, and 7 systems. The Minimum Bandwidth was calculated to two decimal places. The system setup used two nodes which connected together to a third node. All data then traveled in a linear direction for the rest of the nodes. The data was set up for a packet size of 100 bits. The experiment used CBR over UDP and FTP over TCP. Each node used 10ms of lag, with packets being selected to drop in droptail fashion. We noted that there is a change in the number of dropped packets depending upon how many nodes are in our system. However, this change is nearly negligible; as it is so small it can be considered inconsequential. This allows our system to use any number of cars in a group without worrying about losing packets as transmissions are passed. We also came up with results which showed that the number of nodes in our system have an effect on the minimum bandwidth required to have no dropped packets. If we have 4 nodes in our system, we see a lower minimum bandwidth to prevent any packet dropping from occurring. There were a number of limits to how these experiments were tested. The first is a restriction of node usage. NS2 only allows for a maximum of 7 nodes to be used at one time. This means that any group size larger than 7 would be untestable in this fashion. The second is the fact that NS2 is only a simulator. It works on pre-programmed algorithms and cannot truly show how the system will interact. Finally, the test was performed using a wired system, which may perform differently.

To determine how powerful an in-car processor will need to be to handle being the lead car, that is sending and receiving all messages with the cloud and distributing these messages throughout a group. Based on experimental analysis we conclude it is completely possible to both encrypt and decrypt rsa in our proposed vehicular network. If rsa encryption and time the encryption and decryption of data is used the data will vary in length starting at a length of 1 ascii character and ending at 100 in intervals of 5 characters. This will allow us to view how long the encryption and decryption will take based upon the length of our message.

Starting with a rather powerful computer or Lead Car, messages will be sent to the lead car from a computer simulating the tower. These messages will be deciphered using a public key, and the messages will be encoded into cypher using a private key, and sent out to each car in the group. In essence, each car will be a computer connected to the testing computer. These "cars" will decode the message, and reply with a random set of coordinates in cipher text. The lead car will then decipher these messages, encrypt them with its public key, and send them back to the tower. We will start with a group size of one other car, and slowly increase the amount of cars. To increase the amount of cars, the lead car will receive a key from the tower, and the new computer will send initiate a handshake, using the same key. Once handshake is complete,

the lead car will have to distribute this new key to the group. We will monitor the lead cars processor throughout this procedure.

Initial public and private keys, as well as keys for adding additional "cars", coordinates to be sent to the tower, as well as arbitrary instructions to be sent from the tower to the cars.

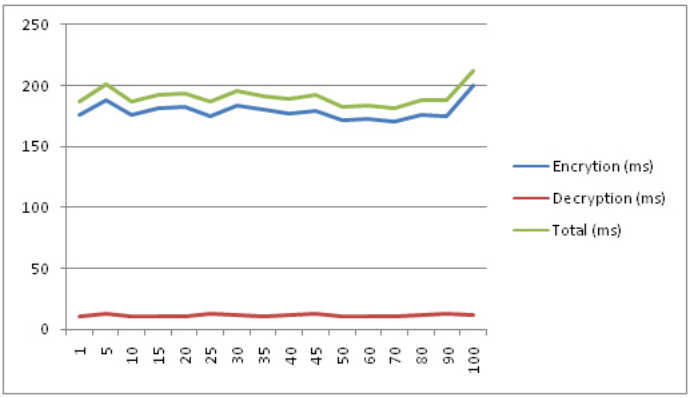


Figure 13

This experiment was performed on a general desktop computer with Pentium i5 processor at a clock speed of 2.53 GHz. The data was retrieved from a java program in eclipse, which took the time before and after encryption and decryption. It was then displayed for processing. This made use of the RSA functionality in the java API. According to these results, we see very little change in the level of encryption and decryption based on the length in characters. This tells us that we can send multiple GPS coordinates in one string without much loss in time. We also note that it take far less time to decrypt RSA than to encrypt, which means that more problems will arise when trying to send a message then receiving one. This means that each car in the group other than the lead car will have far less to process, as the lead car has to send many messages as opposed to one or two.We are limited in a number of ways in this experiment. The computer we used was more powerful than anything that would be considered for this car experiment. This means that any problems that would arise in longer encryption may be overlooked. We are also limited in the length of supported characters we can test. The maximum supported character length is 117 characters. This was recorded, but was left out of the charts. Finally, this shows us what the time it would take to encrypt would be if the processor wasn't trying to do other things as well. This is abstracted compared to the actual problem.

6. CONCLUSION AND FUTURE WORK

In order for V2V technology to be secure, policies and procedures should be put in place to keep hackers and threats at bay. Security needs to be transparent so that clients can see how there info is protected. This includes the SLA. This paper has looked at the effect of group security on V2V, and provided a framework to analyze performance of secure V2V system. In our proposed approach, a cloud is responsible for storing all information and doing most of the computation. Each cell tower is connected to the cloud, and relay information to and from it. Each cell tower connects to a certain amount group of car. Each car in the group is connected to each other car, and the group is connected to the tower through the car with the best connection. In our scheme we concentrate on group security rather than individual security. For future work, it is important the system be able to detect infection or abnormal condition real-time, in order to inform the driver. Self-diagnosis, Self-detection and Self warning are important features. The final suggestion to maintain safety is by tracking the key components more efficiently through

design, such as brakes, doors and the engine. If a car is infected, these will be the most dangerous to tamper with.

REFERENCES

Alexiou, N., Gisdakis, S., Laganà, M., and Papadimitratos, P. 2013. Towards a secure and privacy-preserving multi-service vehicular architecture. In *14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, Madrid, pp.1–6.

Alexiou, N., Laganà, M., Gisdakis, S., Khodaei, M., and Papadimitratos, P. 2013. Vespa: Vehicular security and privacy-preserving architecture. In *2nd ACM workshop on Hot topics on wireless network security and privacy*. ACM, pp.19–24.

Baldini, G., Mahieu, V., Fovino, I. N., Trombetta, A., and Taddeo, M. 2013. Identity-based security systems for vehicular ad-hoc networks. In *International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, pp. 672–678.

Crosby, G. V. and Vafa, F. 2013. Wireless sensor networks and lte-a network convergence. In *38th Conference on Local Computer Networks (LCN)*. IEEE, Sydney, NSW, pp. 731–734.

Leinmuüller, T., Buttyan, L., Hubaux, J. P., Kargl, F., Kroh, R., Papadimitratos, P., Raya, M., and Schoch, E. 2006. Sevecom - secure vehicle communication. In *IST Mobile Summit*.

Liao, C., Chang, J., Lee, I., and Venkatasubramanian, K. K. 2013. A trust model for vehicular network-based incident reports. In *5th International Symposium on Wireless Vehicular Communications (WiVeC)*. IEEE, Dresden, pp. 1–5.

Mangel, T. and Hartenstein, H. 2011. An analysis of data traffic in cellular networks caused by inter-vehicle communication at intersections. In *Intelligent Vehicles Symposium (IV)*. pp. 473–478.

Panja, B., Morrison, D., Turner, S., and Meharia, P. 2014. Integration of v2v with cloud computing to provide group security through the use of 4g lte. In *9th International Conference on Cyber Warfare & Security*. Academic Conferences Limited, pp.167.

Rangarajan, S., Verma, M., Kannan, A., Sharma, A., and Schoen, I. 2012. V2c: a secure vehicle to cloud framework for virtualized and on demand service provisioning. In *International Conference on Advances in Computing, Communications and Informatics*.

Sagstetter, F., Lukasiewycz, M., Steinhorst, S., Wolf, M., Bouard, A., Harris, W. R., Jha, S., Peyrin, T., Poschmann, A., and Chakraborty, S. 2013. Security challenges in automotive hardware/software architecture design. In *Conference on Design, Automation and Test in Europe*. EDA Consortium, pp. 458–463.

Seong-Woo, K. and Seung-Woo, S. 2012. Cooperative unmanned autonomous vehicle control for spatially secure group communications. *Selected Areas in Communications, IEEE Journal 30,* 5 (June), pp.870 – 882.

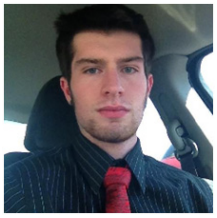
Tsuboi, Tsutomu, and Sekiguchi, T. 2014. Optimization for wireless vehicular network system in urban area. In *Communication Technologies for Vehicles*. Springer International Publishing, pp. 126–142.

Wang, E. K., Ye, Y., and Xu, X. 2014. Location-based distributed group key agreement scheme for vehicular ad hoc network. *International Journal of Distributed Sensor Networks*.

Zhu, M. and Martinez, S. 2012. On resilient consensus against replay attacks in operator-vehicle networks. In *American Control Conference (ACC)*. pp.3553–3558.

Biswajit Panja is an assistant professor of computer science and University of Michigan-Flint. His research is in the areas of scalable sensor networks, mobile ad hoc networks, and network security. He has published over 20 peer reviewed papers and have secured grant funding from NASA and KSTC. He has a PhD in computer science from the University of Missouri-Rolla.

David Morrison is a Computer Science undergraduate major with a concentration in Software Engineering at The University of Michigan-Flint. He lives in Romeo, Michigan. His other interests include Mathematics and Music. He works with Dr. Panja as a UROP.



Priyanka Meharia is an Assistant professor of Accounting Information Systems at Eastern Michigan University. Her research is in the area of Audit, Control and Security of Systems. She has published over 10 papers. She has PhD from University of Kentucky. She is a Certified Public Accountant trained in software such as SAP ERP (Finance and Control), Business Analytics using SAP BI tools, Microsoft Power BI, Audit software-Idea and Ultratax.

Prof. Bharat Bhargava is currently a Professor of computer science at Purdue University. He received the BE degree from the Indian Institute of Science, and the MS and PhD degrees in electrical engineering from Purdue University, West Lafayette, IN. He His research involves mobile wireless networks, secure routing and dealing with malicious hosts, providing security in Service Oriented Architectures, adapting to attacks, and experimental studies. His name has been included in the Book of Great Teachers at Purdue University. Moreover, he was selected by the student chapter of ACM at Purdue University for the Best Teacher Award. He is a fellow of the IEEE.