Securing the Positioning Signals of Autonomous Vehicles

Gabriel Esguerra[1], Kimiya Ghobadi[1], Jimson Huang[1], Robin Hill[2], Derwin Lawson[2], Stone Li[3], Tiffany Zhan[3], Shahab Tayeb[4]
[1]AEOP UNITE, [2]RET, [3]UNLV STEM, [4]UNLV
Las Vegas, Nevada

***Abstract*—One of the fastest growing industries in America is autonomous vehicle technology. The main motivation is to decrease the number of accidents each year. This leads to the challenge presented in this paper, preventing the spoofing of signals coming into autonomous vehicles. The increased complexity of these vehicles creates more vulnerabilities for attackers to take advantage of. Authentication of vehicular ad hoc networks is one method to help stop the potential hacking of autonomous vehicles. We propose a two-factor authentication for GPS signals by synchronizing with Stratum-1 clocks and digital signatures to prevent man-in-the-middle attacks. The experiment uses a computer to simulate a GPS signal that is sent to a Raspberry Pi 3 along with a timestamp and hashed key using RSA-1024. The Raspberry Pi 3 represents the vehicle. The method presented will prevent GPS spoofing attacks that are using modified or corrupted messages, an impersonation attack, or a roadside unit replication attack.**

***Keywords*—Autonomous vehicle, two-factor authentication, time synchronization, digital signature, GPS spoofing**

I. Introduction

A. Purpose

The technology behind autonomous vehicles is one of the fastest growing industries in America. They are disrupting the automobile industry as several companies, such as Google, Tesla, BMW, Toyota, and Mercedes are developing their own autonomous vehicles. The security of these systems could affect the lives of millions in the near future and could become an ethical dilemma. There are multiple reasons why people are switching over to autonomous driving other than convenience. Self-driving cars are cost efficient and provide increased safety compared to human-operated cars.

Safety is the most crucial aspect of any vehicle. According to Urmson et. Al, more than 1.2 million people die and 2.7 million people are injured in traffic accidents each year in the U.S. alone [1]. A vast majority of these accidents are caused by human errors, which could be prevented by autonomous driving. According to Dhar, "even-less-than-perfect (sic) systems could be compelling enough if they substantially reduce accidents, considering that every percentage point of improvement equates to 400 lives saved and 40,000 injuries avoided annually" [2]. As a result, more people are investing in these vehicles and more companies and government entities are working hard to bringing this technology to the public.

The benefits go even beyond safety and cost. The adaptive nature of autonomous vehicles would allow people who are unable to drive, whether due to age or disability, to break through the barrier, creating more customers for the automobile market [3]. The increased number of self-driving vehicles could also reduce driving's environmental impact. Due to its newer technology, most autonomous vehicles are fully electric or hybrid and are more fuel efficient than human-operated cars [1].

B. Motivation

As mentioned above, autonomous vehicles are an upcoming technology that will provide many benefits to our daily life. One of the most critical devices of an autonomous vehicle is GPS. Position spoofing is a potential risk in these brand-new technologies, and it provides a huge attack surface for potential attackers. Position spoofing is when the attacker misleads a GPS using incorrect GPS signals. To launch the positioning attack is easy, but it is difficult to be prevented by the owner of the automobile [4]. The probability of success of an attack is high [4]. This strongly motivates us to conduct research on position spoofing and provide better security and safety for future autonomous vehicle passengers.

C. Hypothesis

Due to the standard of having a positioning system built into autonomous cars, it is important that protection against spoofing be implemented. To ensure that only legitimate signals are accepted to the autonomous vehicle, all incoming signals need to have an accurate timestamp accompanying them. [5]. While this may deter most attacks, more advanced adversaries may have the ability to synchronize their system clock to a more accurate clock. This is why the proposed approach to protecting against position spoofing is two-factor authentication. The vehicle will take advantage of not just time synchronization, but also digital signatures. Self-driving vehicles will need to be synchronized with clocks with higher accuracy than most clocks on the planet such as a low stratum level NTP4 (Network Time Protocol Version 4). If an attacker were to synchronize with a similarly accurate clock, the digital signature would detect their tampering. The hash creates a digital signature that any hacker would have difficulty breaking. With the dangers of positioning systems influencing how an autonomous vehicle maneuvers the streets, it is increasingly important to implement a countermeasure.

D. Background

In 2010 Google announced they had seven autonomous vehicles and had driven over 140,000 miles on California highways [6]. By the fall of 2016 that number was over 2 million miles. In that time, there were 14 accidents, but in only one was the google car at fault [7]. Today most major car manufacturers such as Audi, BMW, GM, Nissan, Toyota and Volvo all have active programs to develop autonomous cars which may save over 2.4 billion gallons of gasoline [6].

As cars become more reliant on computers and sensors to navigate, solving the cyber attacking problem will become an important part of the equation for safer cars [8].

For autonomous vehicles to drive cooperatively they need to share information with each other. VANETs (Vehicular ad hoc networks) are one of the ways this can be accomplished, but VANETs have numerous security issues [8]. A VANET network, also known as a vehicular ad hoc network, is a type of automobile technology that allows vehicles to relay information such as road conditions and travel situations. Computers systems are prone to hacking and the GPS (Global Positioning System) technology is no exception. The satellite signal alone cannot be used to drive the car. Satellite GPS is supplemented with WADGPS (Wide Area Differential Global Positioning System) [9]. GPS and VANETs can all be targets for hackers. Car performance and safety features are easy for car manufacturers to predict, however cyberattacks are difficult to defend against; even worse, security systems for the vehicles are limited by cost and must not interfere with vehicle performance [10].

Spoofing is the transmission of matched-GPS signal structure interference in an attempt to commandeer the tracking loops of a victim receiver and thereby manipulate the receiver's timing or navigation solution [11]. Anti-spoofing techniques exist that would enable receivers to successfully defend themselves against such attacks. These techniques enable a receiver to differentiate authentic GPS signals from counterfeit signals with high likelihood of success. Spoofers have the ability to take control of the victim receiver's tracking loops and lead the receiver away from the authentic signals. There is a point when the spoofed signals have moved more than 600 meters in position or 2 microseconds in time away from the authentic signals, the receiver can be considered completely owned by the spoofer [11]. Wesson classified spoofing attacks into simple, intermediate or sophisticated in terms of their effectiveness and subtlety.

Wesson proposed Spread Spectrum Security Codes, which is a Cryptographic Method. A broadcast GPS signal is sent over a satellite once the receiver receives the legitimate signal the next broadcast can be predicted; however, the exact broadcast sequence cannot be determined. The receiver then stores the broadcast into memory. When the cryptographic digital key is sent again the receiver generates a copy and compares it with the first broadcast key. Spoofing is declared if the correlation power falls below a predetermined threshold [11].

When conducting an intermediate spoofing attack, the spoofer will approximate the location and speed of the target antenna, then the spoofer will send out an imposter signal to the GPS signal [12].

The most common spoofing method is the simplistic spoofing attack this attacker uses a commercial GPS signal simulator to broadcast their signal. In 2002, this test was restructured in the Los Alamos National Laboratory. The disadvantage of using this method is it is very costly, and very heavy to move. Also, they are easily detected and can only jam the authentic signal [12].

Another more advanced spoofing method is the sophisticated spoofing attack. The angle of attack defense is not successful in this spoofing attack, the only successful defense is the cryptographic authentication [12].

Most of the cryptographic authentication techniques require some modifications in the GPS signal structure. Therefore, this method does not seem to be applicable to GPS in short term [13].

In 1999, the FCC (Federal Communications Commission) set aside a portion of the electromagnetic spectrum for DSRC (Dedicated Short Range Communication). The latency is less than 50 milliseconds and the range is about a kilometer and transfer rates range from 3 to 27 Mbps (Megabits per second). This network enables vehicle communications both between vehicles V2V and vehicles and the general infrastructure (V2I) [14].

Abueh and Liu have identified the following threats to VANETs:

- Authentic messages are resent by an attacker to overwhelm the system.
- Fake messages are sent to disrupt the entire network.
- Message modification where messages are modified or corrupted en route to the vehicle.
- An impersonation attack where the attacker pretends to be the actual network.
- Denial of service attack where a vehicle is not allowed on the network.
- RSU replication attack.

The need for security in VANETs must be balanced with the need for real-time decision making in the vehicles and the large number of vehicles that may soon be on the network. Vehicles communicate with the DSRC every 100 – 300 microseconds, and if there are hundreds of vehicles, cars could be verifying several thousand messages every second [15].

Wang and Wang wrote that two-factor authentication schemes have been proposed by many authors, the results have only complicated the process without any real security improvements [16].

In this paper, we propose using the GPS time stamp along with the hashed RSA-1024 coded message. This will eliminate the possibility of GPS spoofing on autonomous vehicles because the GPS Stratum-2 timestamp will be used to determine if messages are coming from the network and if there is a hashed certificate. So, although two-factor security has been criticized in the past, we hope to show the potential for using time to detect non-authentic messages from being

used by autonomous vehicles and all vehicles using GPS navigation.

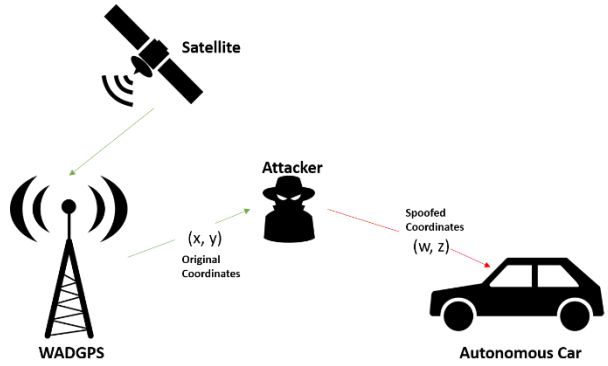

Figure 1. Visual representation of man-in-the-middle attack

In Figure 1, the attacker intercepts a GPS signal and modifies the coordinates being sent to the car. This causes a self-driving car to reroute to a potentially dangerous area.

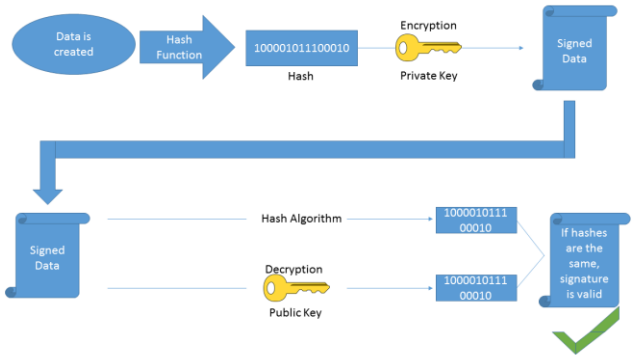

Figure 2. Generic flowchart of a digital signature using PKI

A digital signature is a method of securely connecting a signer and a document that uses Public Key Infrastructure (PKI), a set of requirements that allows digital signatures to generate a private and a public key. The private key is used to "sign" the document and the public key is used to verify the signature. Digital signatures are often used when there is a need to validate and authenticate electronic messages exchanged over a network [17]. According to Aki, it guarantees that no one can forge a sender's signature [17]. Figure 2 depicts the process.

II. System Model

A. Network Model

A standard VANET scenario was used. The scenario consisted of a network with a communication tower or Roadside Unit (RSU), and an autonomous vehicle [15]. The RSU handled most of the computational tasks involved with digital signatures such as public and private key generation as well as hashing. An RSU was placed along the roadside in a central location to ensure broad coverage. GPS satellites will know the exact position of the RSU which will allow it to calculate a car's position by using itself as a reference point. The RSU was synchronized with a Stratum-1 time server thus becoming a Stratum-2 device. Each autonomous vehicle was equipped with a GPS receiver that is capable of small computational abilities such as decryption, hash comparison, and time comparison. Every receiver was able to synchronize their clocks with RSU clocks. The vehicle in this scenario received data from the RSU but did not send any data out.

B. Adversary Model

The adversary can easily spoof unauthenticated GPS receivers. They are capable of intercepting signals, reading them, altering them, and delaying them. Due to autonomous vehicles relying heavily on GPS for navigation, they are susceptible to pathing attacks that could cause accidents or even casualties. For example, an attacker that intends to terrorize and harm many pedestrians could take control of an autonomous vehicle by tricking its receiver into believing that the sidewalk is a road that leads to the desired destination. This could result in the car driving into a crowd of people harming and possibly killing them. An attacker with intentions to rob the driver could also deceive a car into believing it has reached its destination in a secluded area and rob the driver for their personal gain. The increasing complexity of cars opens more doors for attackers to enter through and cause chaos.

C. Design Goals

In order to secure the VANET, there needs to be a series of authentication methods to prevent an attacker from gaining control of an autonomous vehicle put into place. The first layer of authentication was time synchronization. The second layer was a digital signature. These methods provide several basic security goals for wireless communications.

1. *Prevention of modifying or forgery of the message*

Every message needs to be authenticated to ensure that the data is actually coming from a legitimate source rather than an intruder. Using digital signatures and time synchronization, modified data will be detected and rejected.

2. *Nonrepudiation*

This means that a source cannot deny the authenticity of their signature or the fact that they sent a message. This ensures that even if an outside party were to gain the public key, they cannot generate a legitimate signature. Data is guaranteed to come from a legitimate source.

3. *Keys cannot be reused or moved to another document*

If an attacker were able to capture a public key, they will not be able to reuse it. The RSU generates a new public key for every signal sent out. Furthermore, the digital signature becomes part of the document itself and cannot be moved to another set of data.

The primary function of a digital signature is to ensure that data has not been modified or intercepted and replaced.

$$f(x) = y$$

The process can be represented by a series of variables. X represents the original data while f() is the hashing algorithm and y is the outcome of the hash

Keep in mind that f(x) is a one-way mathematical function that is almost impossible to reverse. The function is similar to multiplying two large prime numbers that result in a large non-prime number. Due to the sheer amount of combinations that can be multiplied to get that specific number, it would take longer than the lifespan of the universe to find the two factors used. The hash is then encrypted using another function with the sender's private key. This results in a signed message that is sent to the client with a public key.

The encryption by the private key can be represented by a function g() which takes the hash, y as a parameter. g(y) is equal to z which is sent to the car along with another function

$$g(y) = z$$
$$h(z) = y$$

h(). Think of h() as the public key or inverse function of g(). It takes the encrypted hash, z, as the parameter. If the data was not changed during mid transmission, then the decrypted hash should match the original hash.

```
if h(z) == y {
      Data is valid
      Accept data
}
else {
     reject data
}
```

Figure 3. Pseudocode for securing the signal

The pseudocode in Figure 3 shows a generic if-else statement that depicts the decision made by the car depending on if the hashes match or not. If the hashes are not equal, that means the data has been tampered with in some way. This would result in the car rejecting the data thus nullifying the attempt to spoof.

Time synchronization functions as a preliminary filter for all GPS signals. Cars need to be able to make decisions quickly. To combat congestion and danger at high speeds, there needs to be a way to reduce the computation load. Before going into the verification phase of the digital signature, there will be an initialization phase that determines whether or not the car will even try to verify the signature. When an RSU creates data, it will send the data with a timestamp. The proposed infrastructure requires that the RSU be connected to a Stratum-1 server running NTP version 4, effectively becoming a Stratum-2 device. A Stratum-1 clock is essentially as accurate as possible without being the reference atomic clocks known as Stratum-0 clocks. Then, the car itself will have its clock synchronized with the RSU clock which turns the car into Stratum-3. This degree of accuracy is needed to ensure that attackers cannot send a signal that can deceive the vehicle even with a higher signal strength than an RSU. The reasoning behind this is that clocks drift off the reference clocks overtime. Most attackers using a computer will be using a system clock which is present on every computer. NTP servers tend to be accurate within milliseconds meaning that most attackers will not be able to match this accuracy. Even if the attacker intercepts a signal and changes the data, the timestamp changes with the data.

Figure 4. Diagram of time synchronization

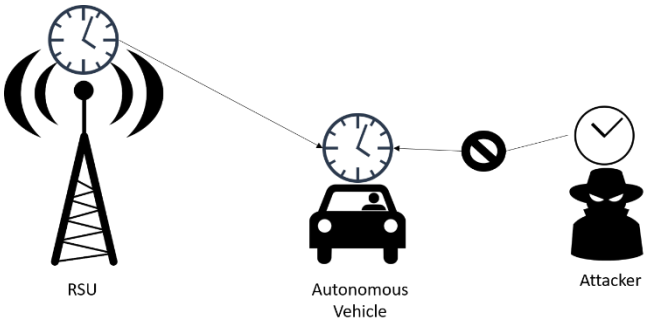

Figure 4 shows how the RSU and autonomous vehicle have identical clocks whereas the attacker's clock is not as precise. The car compares the timestamp of the data to the current time to see if it was received within a reasonable time. If the timestamp is off by too much, the data has been tampered with and should not be used, otherwise, the data is valid. In order to detect tampered data and accept genuine data, a threshold needs to be created that is accurate but not so accurate that RSU signals are rejected. By implementing this structure, man-in-the-middle attacks are rendered impossible for most adversaries.

III. Methods

Materials:

For our experiment, we used a Raspberry Pi 3 as the receiver and a computer as the sender. We chose a Raspberry Pi to be the receiver because it was simple and easy to work with. It also works with Python, a programming language which has many encryption libraries. For encryption, we used RSA 1024, 1536, 2048, 2560, 3072, 3584, 4096, 4608, 5120, 5632, 6144, 6656, 7168, 7680, and 8192. To synchronize the clocks, a Stratum-1 time server along with NTP version 4 was used. The Python library PyCrypto provided the RSA algorithms, while the Python library ntplib was used to query NTP servers.

Process:

Initially, the computer simulating an RSU is connected to a Stratum-1 clock. Then, it sends the original signal, the time, the public key, and the hashed signal to the car. Next, the Raspberry Pi 3, which simulated the car, tests if the time is in

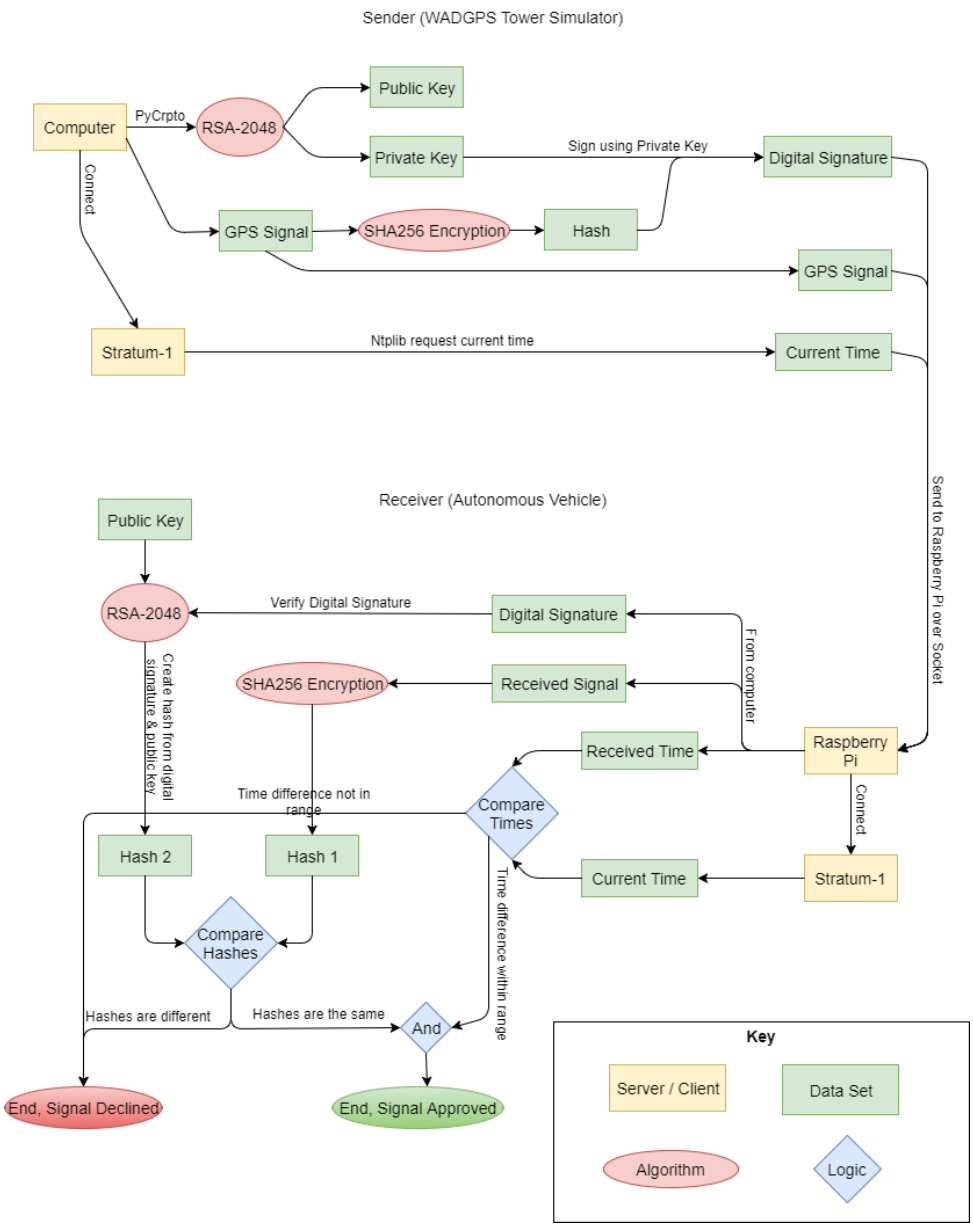

Figure 5. Flowchart of the process of our method

the allowed time frame. We used NTP4 to sync the time in the two networks. If the signal is not received in the specified time range, then the car rejects the signal, proving that the signal is false or has been modified. If the signal is within the acceptable time range, then the car verifies the digital signature. The car hashes the signal using the public key, which was made using RSA-1024, in order to create another signature. Then it checks if the signature is the same as the one received. If it is not, the car rejects the signal and the process repeats until a signal is approved to be safe and can be used. We used RSA-1024 because it is uncrackable due to the amount of resources required to brute force it [18]. Figure 5 provides a graphical representation of the process. A detailed description of our programs follows:

Transmitter (Computer)

1. A socket server is created on the computer's IP for transmitting data.

2. Using the Python library PyCrypto, a public and private key with a size of 2048 bits are created, and the public key is broadcast to all receivers. We used a key size of 2040 bits as it is currently uncrackable. A higher number of bits would be more secure but at a cost of longer creation, signing, and verification times which could cause problems for vehicles moving at high speeds. Figure 5 shows the relationship between the time taken to sign and verify the digital signatures in seconds to the RSA key sizes in bits. This is a potential problem because autonomous cars need to make

decisions quickly. Although the hash is harder to crack, speed is sacrificed.

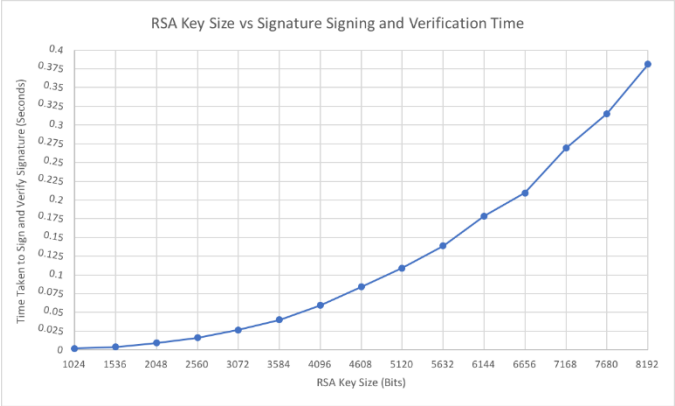

Figure 6. Graph showing decrease in speed with increasing key size

3. The receiver connects to the server.

4. The GPS signal is hashed using SHA256 and then signed using the private key to create the digital signature.

5. The computer connects to a Stratum-1 server through NTP using ntplib, another Python library.

6. The computer requests the current time from the Stratum-1 server.

7. The computer sends the GPS signal, the digital signature, and the current time, all encoded as UTF-8, to the Raspberry Pi.

Receiver (Raspberry Pi)

1. The Raspberry Pi establishes connection to the computer using a socket client.

2. The Raspberry Pi receives the GPS signal, the digital signature, and the time the signal was sent from the computer, and decodes them as UTF-8.

3. The Pi connects to the same Stratum-1 server through NTP using ntplib.

4. This is the first step in the two-factor authentication. The time of transmission is checked against the current time. If the time gap is more than the time it takes to receive the signal, calculated using the distance between the receiver and the transmitter, plus 50 milliseconds of leniency, this would indicate a man-in-the-middle attack. If the time difference does fall within range, the second step in the two-factor authentication, checking the digital signature, begins.

5. The Pi decrypts the digital signature using the transmitting server's public key to generate the hash of the original signal.

6. To verify that the received signal was not tampered with, the Pi creates a hash of the received signal with PyCrypto using the same SHA256 algorithm that the transmitter used.

7. If the two hashes match up, the signal is accepted. If the two hashes do not match up, the signal is voided.

8. If the signal is voided in the first or second step of the authentication process, a request for a new signal is sent to the transmitter.

IV. Results and Discussion

After two-factor authentication was implemented on both the computer and the Raspberry Pi, the Raspberry Pi was able to receive signals and check its integrity by using a time stamp and a digital signature received from the computer.

The screenshots below show the results of two tests. For the test in Figure 8, the GPS signal was changed by a packet sniffing program. The test simulated a real-life example of a man-in-the-middle attack where a hacker tries to hijack the GPS signal. For these tests, we used a time leniency of 300 ms to simulate a real-life distance between the car and the WADGPS tower. In Figure 7, the receiver detects that the received time and current time are within the 300ms difference and the digital signature can be verified by the received signal. In Figure 8, however, the receiver detects too long of a time gap (315 ms) and the signal cannot be verified by the digital signature.

```
pi@raspberrypi:~/Desktop $ python3 packet_receiver.py
Thank you for connecting
Received Time: 1500325674.1467586
Current Time: 1500325674.1577668
Received message on time!
Sinature Received: 66423546903...
Digital signature successfully verified!
pi@raspberrypi:~/Desktop $
```

Figure 7. Console output on the Pi of a successful verification of the time stamp and digital signature

```
pi@raspberrypi:~/Desktop $ python3 packet_receiver.py
Thank you for connecting
Received Time: 1500325951.6226773
Current Time: 1500325951.9377089
Warning! Message was delayed!
Sinature Received: 66423546903...
Warning! Digital signature could not be verified!
pi@raspberrypi:~/Desktop $
```

Figure 8. Console output on the Pi of an unsuccessful verification of the time stamp and digital signature

We implemented a method for securing the GPS signals in autonomous vehicles. Our proposed approach was a two-factor authentication method using time synchronization and a digital signature. Prevention of GPS spoofing is critical due to the way these vehicles navigate. By detecting this

modification, the car can reject falsified data. A different two-factor authentication scheme that uses certificate authority and a biological password was also successful and shown to be efficient [15]. This means that two-factor authentication for an autonomous vehicle is feasible, and may also mean that multi-factor authentication could be used. Due to time constraints, however, we were unable to fully test other methods of authentication.

This research provides a solution to GPS spoofing in autonomous vehicles which ensures the safety of the passengers. In the future, a more secure authentication method may need to be implemented to prevent hackers from hijacking the car.

V. Conclusion and Future Works

In this paper, we tested a two-factor authentication method to prevent the GPS spoofing of autonomous vehicles. Of the six previously mentioned VANET threats, three of them have been solved. We successfully prevented acceptance of message modification, impersonated signals, and identified RSU replication attacks. GPS spoofing is a major issue for cars, especially for autonomous vehicles because of the way some of them navigate. Our method used a digital signature and time synchronization. We created a digital signature using RSA-1024 to hash the GPS signal, then encrypted the hashed signal using a private key. The car decrypts the hashed and encrypted signal using the public key that was also sent.

Although our method of two-factor authentication was successful, more layers may be required in the future as technology evolves. An example of another security layer that could be added is VIN number authentication. In a VANET, vehicles may communicate with each other and to ensure validity of communications, each signal must be accompanied by a VIN number. Implementing biometrics would further help the security of vehicles without sacrificing speed as it takes place before signature verification.

Acknowledgment

This material is based upon work supported in part by the Department of Defense under Army Educational Outreach Program (AEOP) and the National Science Foundation under Grant No. 1710716. The authors thank the UNLV writing center for helping with revising the manuscript.

VI. References

[1] Urmson, C. and Whittaker, W. (2008). Self-Driving Cars and the Urban Challenge. *IEEE Intelligent Systems*, 23(2), pp.66-68.

[2] Dhar, V. (2016). Equity, Safety, and Privacy in the Autonomous Vehicle Era. *Computer*, 49(11), pp.80-83.

[3] Bourhis, G., Horn, O., Habert, O. and Pruski, A. (2001). An autonomous vehicle for people with motor disabilities. IEEE Robotics & Automation Magazine, 8(1), pp.20-28.

[4] Petit, J. and Shladover, S. (2015). Potential Cyberattacks on Automated Vehicles. IEEE Transactions on Intelligent Transportation Systems, pp.1-11.

[5] Mundt, T. (2006). Two methods of authenticated positioning. In *Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks* (Q2SWinet '06). ACM, New York, NY, USA, pp. 25-32.

[6] Tannert, C. 2014. Self-Driving Cars: Inside the Road Revolution. . http://www.fastcompany.com/3022489/innovation-agents/self-driving-cars-let-go-of-the-wheel

[7] Bhuiyan, J. (2016). After two million miles, Google's robot car now drives better than a 16-year-old. [online] Recode. Available at: https://www.recode.net/2016/10/5/13167364/google-self-driving-cars-2-million-miles

[8] Amoozadeh, M., Raghuramu, A., Chuah, C., Ghosal, D., Zhang, H., Rowe, J. and Levitt, K. (2015). Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Communications Magazine*, 53(6), pp.126-132.

[9] Kavitha J. C., Venugopal, A., Pushparani, S. (2016). Algorithm for Security in Autonomous Cars. International Conference on Computer Applications, pp. 83-88.

[10] H. Yu and C. W. Lin, "Security concerns for automotive communication and software architecture," (*2016) IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, San Francisco, CA, 2016, pp. 600-603.

[11] Wensson, K. Shepard, D., Humphreys, T. (2012). Straight Talk on Anti-Spoofing. GPS World 23(1), pp. 32-39.

[12] Haider, Z., & Khalid, S. (2016). Survey on effective GPS spoofing countermeasures. *In Innovative Computing Technology (INTECH), 2016 Sixth International Conference on* (pp. 573-577). IEEE.

[13] Jafarnia-Jahromi, A., Broumandan, A., Nielsen, J. and Lachapelle, G. (2012). GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques. *International Journal of Navigation and Observation*, 2012, pp.1-16.

[14] Y. J. Abueh and H. Liu, "Message authentication in driverless cars," (*2016) IEEE Symposium on Technologies for Homeland Security (HST)*, Waltham, MA, 2016, pp. 1-6.

[15] Wang, F., Xu, Y., Zhang, H., Zhang, Y. and Zhu, L. (2016). 2FLIP: A Two-Factor Lightweight Privacy-Preserving Authentication Scheme for VANET. *IEEE Transactions on Vehicular Technology*, 65(2), pp.896-911.

[16] Wang, D. and Wang, P. (2016). Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound. *IEEE Transactions on Dependable and Secure Computing*, pp.1-1

[17] Aki, S. G (1983). Digital signatures: A tutorial survey. *Computer*, 16(2), pp.15-24.

[18] Bos, J., Kaihara, M., Kleinjung, T., Lenstra, A. and Montgomery, P. (2009). *On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography*. [ebook] Lausanne: Cryptology ePrint Archive, Report 2009/389. Available at: https://infoscience.epfl.ch/record/164549