Detecting Sybil Attacks using Proofs of Work and Location in VANETs

Sanjeevini S.H[1], L.Savithri[2], E.Maha Lakshmi[3], M.Amulya[4], P Sanjana[5]
[1]Assistant Professor, Department of CSE(CS), MallaReddy Engineering College for Women, Hyderabad, TS, India.
Email: sanjeevinicsecs@gmail.com
[2,3,4,5]UG Students, Department of CSE(CS), MallaReddy Engineering College for Women, Hyderabad, TS, India.

ABSTRACT
Vehicular Ad Hoc Networks (VANETs) has the potential to enable the next-generation Intelligent Transportation Systems (ITS). In ITS, data contributed from vehicles can build a spatiotemporal view of traffic statistics, which can consequently improve road safety and reduce slow traffic and jams. To preserve vehicles' privacy, vehicles should use multiple pseudonyms instead of only one identity. However, vehicles may exploit this abundance of pseudonyms and launch Sybil attacks by pretending to be multiple vehicles. Then, these Sybil (or fake) vehicles report false data, e.g., to create fake congestion or pollute traffic management data. In this paper, we propose a Sybil attack detection scheme using proofs of work and location. The idea is that each road side unit (RSU) issues a signed time-stamped tag as a proof for the vehicle's anonymous location. Proofs sent from multiple consecutive RSUs is used to create vehicle trajectory which is used as vehicle anonymous identity. Also, one RSU is not able to issue trajectories for vehicles, rather the contributions of several RSUs are needed. By this way, attackers need to compromise an infeasible number of RSUs to create fake trajectories. Moreover, upon receiving the proof of location from an RSU, the vehicle should solve a computational puzzle by running proof of work (PoW) algorithm. So, it should provide a valid solution (proof of work) to the next RSU before it can obtain a proof of location. Using the PoW can prevent the vehicles from creating multiple trajectories in case of low-dense RSUs. Then, during any reported event, e.g., road congestion, the event manager uses a matching technique to identify the trajectories sent from Sybil vehicles. The scheme depends on the fact that the Sybil trajectories are bounded physically to one vehicle; therefore, their trajectories should overlap. Extensive experiments and simulations demonstrate that our scheme achieves high detection rate to Sybil attacks with low false negative and acceptable communication and computation overhead
.

INTRODUCTION
Over the last two decades, Vehicular Ad Hoc Networks (VANETs) have been emerging as a cornerstone to the next generation Intelligent Transportation Systems (ITSs), contributing to safer and more efficient roads. In VANETs, moving vehicles are enabled to communicate with each other via intervehicle communications as well as with road-side units (RSUs) in vicinity via RSU-to-vehicle communications. As a result, a wide spectrum of applications have been emerged as promising solutions [1] to enable new forms of ubiquitous traffic management applications that are not possible with our current traditional

transportation system. The core idea of these applications is to enable vehicles to contribute with data and feedback to an event manager which can build a spatiotemporal view of the traffic state and also to extract important jam statistics [2]. These applications have the potential to contribute to safer and more efficient roads by enabling a wide range of applications such as pre-crash sensing and warning, traffic flow control, local hazard notification, and enhanced route guidance and navigation [3].

However, the aforementioned applications depend on information sent from participating vehicles. Therefore, it is required to preserve drivers privacy especially location privacy while still verifying their identities in an anonymous manner [4], [5]. A naive solution is to allow each vehicle to have a list of pseudonyms to be authenticated anonymously. However, a malicious vehicle may abuse this privacy protection to launch Sybil attack [6]. In Sybil attacks, a malicious vehicle uses its pseudonyms to pretend as multiple fake (or Sybil) nodes [7]. The consequences of a Sybil attack in VANETs can be disastrous. For example, a malicious vehicle can launch the attack to create an illusion of traffic congestion. Consequently, other vehicles will choose an alternative route and evacuate the road for the malicious vehicle. Another potential consequence of a Sybil attack is in safety-related applications such as collision avoidance and hazard warnings where a Sybil attack can lead to biased results that may result in car accidents [3]. Hence, it is of great importance to detect Sybil attacks in VANETs.

Existing works of detecting Sybil attacks can be categorized into three categories, namely, identity registration, position verification and trajectory-based approaches. The ultimate goal of these detection mechanisms is to ensure each physical node is bounded with a valid unique identity. Firstly, identity registration approaches [7–9] require a dedicated vehicular public key infrastructure to certify individual vehicles with multiple pseudonyms to ensure each physical node is bounded with a valid unique identity. However, identity registration alone cannot prevent Sybil attacks, because a malicious node may get multiple identities by non-technical means such as stealing or even collusion between vehicles [10]. Secondly, position verification approaches depend on the fact that individual vehicle can present at only one location at a time. In [11], [3], localization techniques such as Global Positioning System (GPS) are used to provide location information of vehicles to detect Sybil nodes. However, these schemes fail due to the highly mobile context of vehicular networks [12]. Thirdly, trajectory-based approaches is based on the fact that individual vehicles move independently, and therefore they should travel along different routes. In [4], the vehicle obtains its trajectory by combining a consecutive tags from RSUs which it encounters. However, the scheme suffer RSU compromise attack in which if one RSU is compromised, a malicious vehicle can obtain infinite number of valid trajectories. Moreover, in case of rural areas (RSUs are not dense), attackers can create valid trajectories that look for different vehicles.

In this paper, we propose a novel Sybil attack detection scheme using proofs of work and location. The main idea is that when a vehicle encounters an RSU, the RSU should issue authorized time-stamped tag which is a concatenation of time of appearance and anonymous location tag of that RSU. As the vehicle keeps moving, it creates its trajectory by combining a set of consecutive authorized time-stamped tags that are chronologically chained to each other. That trajectory is used as an anonymous identity of the vehicle. Since RSUs have the main responsibility to issue proof of location to vehicles, the scheme should resist against RSU compromise attack so we design the trajectory so that not only one RSU is capable of creating trajectories for the vehicles. To achieve this, threshold signature is adopted so that each RSU is only able to generate a partial signature on a set of time-stamped tags. Once a vehicle travels along a certain threshold number of RSUs, a standard signature representing a proof of location can be generated. Upon receiving an authorized message from an RSU, the vehicle should use it as a seed to solve a puzzle using a proof-of-work algorithm, similar to the one used in Bitcoin [13]. The core idea of POW is to provide a proof to RSUs so they can ensure that the vehicle solved the puzzle correctly. Comparing to Footprint [4], using POW limits the ability of a malicious vehicles to create multiple trajectories.

To detect Sybil trajectories, upon receiving an event from other vehicles, the event manager first applies a set of heuristics to construct a connected graph of Sybil nodes, then it uses the maximum clique algorithm [14] to detect all Sybil nodes in
that graph.

Our main contributions and the challenges the paper aims to address can be summarized as follows:

_ We used threshold signatures to resist RSU compromise attacks. The attacker needs to compromise an infeasible number of RSUs to be able to create fake trajectories.

_ We used the POW algorithm to limit the ability of a malicious vehicle to create multiple forged trajectories, and more importantly, to reduce the detection time for detecting Sybil trajectories which is a critical concern in traffic management applications.

_ We carefully analyzed the probabilistic nature of POW based scheme by examining the affecting parameters (e.g travel time between two consecutive RSUs) experimentally, and then we developed a mathematical model that can be used for adjusting these parameters so that the ability of a malicious vehicle to create forged trajectories is reduced significantly.

_ By experiments, we prove that using the proof of work algorithm reduces the ability of a malicious vehicle to maintain actual multiple trajectories simultaneously. Further simulations, analysis, and practical experiments are conducted to evaluate the proposed scheme and compare it with the Footprint [4], the results indicate that the proposed scheme can successfully detect and defend against Sybil attacks in VANETs and more efficiently compared to the Footprint.

The rest of the paper is organized as follows. We describe the network and threat models in VANETs, followed by the design goal of our Sybil detection scheme in Section II. In Section III, we discuss preliminaries used by this research work. Then, our proposed scheme is presented in Section IV. In Section V, we show the selection of POW parameters values experimentally, and also we provide a mathematical proof of the experimental results. Detailed security and performance
evaluations are provided in Section VI. We present the computation complexity analysis of

our scheme in Section VII. Section VIII discusses the previous research work in Sybil detection in VANETs. Finally, we give concluding remarks in
Section IX.

Exisiting System

Zhou et al. [8] proposed a privacy-preserving scheme based on certificates to detect Sybil nodes. The department of motor vehicle (DMV) represents the certificate authority, and is responsible for providing vehicles with a pool of pseudonyms to be used to hide the vehicle's unique identity. The pseudonyms associated with each vehicle are hashed to a common value. An RSU determines whether the pseudonyms come from the same pool by calculating the hashed values of the received pseudonyms. RSUs can detect Sybil nodes and then report such suspected vehicles to DMV.

To resist against RSU compromise, the paper suggests twolevel hash functions with different keys (coarse-grained keys and fine-grained keys). RSU holds each valid coarse-grained key only for a short time which does not know whether the pseudonyms belong to one vehicle or not. If an RSU is compromised, the attacker only gets the coarse-grained hash key for the current time interval while DMV stores all keys and can detect Sybil nodes by two-level hashing. Although deploying trusted certificates is the most efficient approach that can completely eliminate Sybil attacks, it also violates both anonymity and location privacy of entities. Also, relying on a centralized authority to ensure each is assigned exactly one identity which becomes a bottleneck in the large-scale network such as VANETs.

In [30], Chen et al. proposed a group signature-based approach that can be used to enable a member in the group to authenticate himself/ herself anonymously. Meanwhile, if a particular node generates multiple signatures on the same message, the verifier can recognize those signatures. As a result, detecting duplicated signatures signed by the same vehicles can eliminate Sybil attack. However, the malicious vehicle can launch Sybil attack, if he can generate different messages with similar meaning. Recently, Reddy et al. [7] proposed a cryptographic digital signature based method to establish the trust relationship among participating entities.

The most relevant approach to our work is using trajectories of vehicles as its identities to ensure trust between participating nodes. In [32], RSUs broadcasts digital signatures with a timestamp to vehicles which are under its coverage. Vehicles store the RSUs signatures which they gathered in motion. However, since the time stamp is not issued for a dedicated vehicle, a malicious vehicle may claim its presence at certain RSU by merely eavesdropping such broadcasted timestamp on a wireless channel although it may have never been there at that time. In [4], Footprint has been introduced to detect Sybil attack. When a vehicle passes by an RSU, it obtains a signed message as proof of presence at this location at a particular time. A trajectory of a vehicle is a consecutive series of authorized messages collected by the vehicle as it keeps traveling. Sybil attack can be detected using the fact that the trajectories generated by an attacker are very similar. However, Footprint has some critical issues.

Disadvantages

- The system is not implemented Hashing Keys in order to find Sybil attacks.

- ❖ The system is not implemented attack resistance techniques in order to resist the Sybil and DDOS attacks.

Proposed System

In this paper, we propose a novel Sybil attack detection scheme using proofs of work and location. The main idea is that when a vehicle encounters an RSU, the RSU should issue authorized time-stamped tag which is a concatenation of time of appearance and anonymous location tag of that RSU. As the vehicle keeps moving, it creates its trajectory by combining a set of consecutive authorized time-stamped tags that are chronologically chained to each other. That trajectory is used as an anonymous identity of the vehicle. Since RSUs have the main responsibility to issue proof of location to vehicles, the scheme should resist against RSU compromise attack so we design the trajectory so that not only one RSU is capable of creating trajectories for the vehicles. To achieve this, threshold signature is adopted so that each RSU is only able to generate a partial signature on a set of time-stamped tags. Once a vehicle travels along a certain threshold number of RSUs, a standard signature representing a proof of location can be generated. Upon receiving an authorized message from an RSU, the vehicle should use it as a seed to solve a puzzle using a proof-of-work algorithm, similar to the one used in Bitcoin [13]. The core idea of PoW is to provide a proof to RSUs so they can ensure that the vehicle solved the puzzle correctly. Comparing to Footprint [4], using PoW limits the ability of a malicious vehicles to create multiple trajectories.

To detect Sybil trajectories, upon receiving an event from other vehicles, the event manager first applies a set of heuristics to construct a connected graph of Sybil nodes, then it uses the maximum clique algorithm [14] to detect all Sybil nodes in that graph.

Advantages

_ The system used threshold signatures to resist RSU compromise attacks. The attacker needs to compromise an infeasible number of RSUs to be able to create fake trajectories.

_ The system used the PoW algorithm with Machine learning classifiers to limit the ability of a malicious vehicle to create multiple forged trajectories, and more importantly, to reduce the detection time for detecting Sybil trajectories which is a critical concern in traffic management applications.

_ The system carefully analyzed the probabilistic nature of PoW based scheme by examining the affecting parameters (e.g travel time between two consecutive RSUs) experimentally, and then we developed a mathematical model that can be used for adjusting these parameters so that the ability of a malicious vehicle to create forged trajectories is reduced significantly.

_ By experiments, we prove that using the proof of work algorithm reduces the ability of a malicious vehicle to maintain actual multiple trajectories simultaneously. Further simulations, analysis, and practical experiments are conducted to evaluate the proposed scheme and compare it with the Footprint [4], the results indicate that the proposed scheme can successfully detect and defend against Sybil attacks in VANETs and more efficiently compared to the Footprint.

SModules

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as
Login, Browse and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction Of Attack Status, View Attack Status Ratio, Download Trained Data Sets, View Attack Status Ratio Results, View All Remote Users.

View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT ATTACK STATUS TYPE, VIEW YOUR PROFILE.

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class
Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-Nearest Neighbors (KNN)

- **Simple, but a very powerful classification algorithm**
- **Classifies based on a similarity measure**

- **Non-parametric**
- **Lazy learning**
- **Does not "learn" until the test example is given**

- **Whenever we have a new data to classify, we find its K-nearest neighbors from the training data**

Example

- **Training dataset consists of k-closest examples in feature space**
- **Feature space means, space with categorization variables (non-metric variables)**
- **Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset**

<u>Logistic regression Classifiers</u>

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

<u>Naïve Bayes</u>

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (**Weka 3.6.0**, **R 2.9.2**, **Knime 2.1.1**, **Orange 2.0b** and **RapidMiner 4.6.0)**. We try above all to understand the obtained results.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point *x* and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

Results:

CONCLUSION

Sybil attacks can cause disastrous consequences in VANETS. In this paper, we have introduced a novel approach for detecting Sybil attacks using proofs of work and location. An anonymous trajectory of a vehicle is formed by obtaining a consecutive proof of locations from multiple RSUs which Sybil attacks can cause disastrous consequences in VANETS. In this paper, we have introduced a novel approach for detecting Sybil attacks using proofs of work and location. An anonymous trajectory of a vehicle is formed by obtaining a consecutive proof of locations from multiple RSUs which it encounters. Instead of allowing only one RSU to issue authorized messages for vehicles, at least t RSUs are required for creating a proof of location message using threshold signature to mitigate the RSU compromise attack. Also, the use of proof-of-work algorithm can limit the ability of malicious vehicles to create forged trajectories. Our evaluations have demonstrated that our scheme can detect Sybil attacks with high rate and low false negative rate. Moreover, the communication and computation overhead of the exchanged packets are acceptable.

REFERENCES

[1] F.-J. Wu and H. B. Lim, “Urbanmobilitysense: A user-centric participatory sensing system for transportation activity surveys,” IEEE Sensors Journal, vol. 14, no. 12, pp. 4165–4174, 2014.

[2] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, “Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification,” ACM Transactions on Sensor Networks (TOSN), vol. 11, no. 4, p. 55, 2015.

[3] K. Rabieh, M. M. Mahmoud, T. N. Guo, and M. Younis, “Cross-layer scheme for detecting large-scale colluding sybil attack in vanets,” in 2015 IEEE International Conference on Communications (ICC). IEEE, 2015, pp. 7298–7303.

[4] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen, “Footprint: Detecting sybil attacks in urban vehicular networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, pp. 1103–1114, 2012.

[5] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, “V2x access

technologies: Regulation, research, and remaining challenges," IEEE
Communications Surveys & Tutorials, vol. 20, no. 3, pp. 1858–1877,
2018.

[6] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, "A security and privacy review
of vanets," IEEE Transactions on Intelligent Transportation Systems,
vol. 16, no. 6, pp. 2985–2996, 2015.

[7] D. S. Reddy, V. Bapuji, A. Govardhan, and S. Sarma, "Sybil attack
detection technique using session key certificate in vehicular ad hoc
networks," in Algorithms, Methodology, Models and Applications in
Emerging Technologies (ICAMMAET), 2017 International Conference
on. IEEE, 2017, pp. 1–5.

[8] T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "P2dapsybil
attacks detection in vehicular ad hoc networks," IEEE journal on
selected areas in communications, vol. 29, no. 3, pp. 582–594, 2011.

[9] K. El Defrawy and G. Tsudik, "Privacy-preserving location-based ondemand
routing in manets," IEEE journal on selected areas in communications,
vol. 29, no. 10, pp. 1926–1934, 2011.

[10] Y. Yao, B. Xiao, G. Wu, X. Liu, Z. Yu, K. Zhang, and X. Zhou, "Multichannel
based sybil attack detection in vehicular ad hoc networks using
rssi," IEEE Transactions on Mobile Computing, 2018.

[11] M. S. Bouassida, G. Guette, M. Shawky, and B. Ducourthial, "Sybil
nodes detection based on received signal strength variations within
vanet." IJ Network Security, vol. 9, no. 1, pp. 22–33, 2009.

[12] S. Syed and M. E. Cannon, "Fuzzy logic-based map matching algorithm
for vehicle navigation system in urban canyons," in ION National
Technical Meeting, San Diego, CA, vol. 1, 2004, pp. 26–28.

[13] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[14] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki, "A
simple and faster branch-and-bound algorithm for finding a maximum
clique," in International Workshop on Algorithms and Computation.
Springer, 2010, pp. 191–203.

[15] M. Alsabaan, W. Alasmary, A. Albasir, and K. Naik, "Vehicular networks
for a greener environment: A survey." IEEE Communications
Surveys and Tutorials, vol. 15, no. 3, pp. 1372–1388, 2013.

[16] A. Shamir, "How to share a secret," Communications of the ACM,
vol. 22, no. 11, pp. 612–613, 1979.

[17] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures
based on the gap-diffie-hellman-group signature scheme," in
International Workshop on Public Key Cryptography. Springer, 2003,
pp. 31–46.

[18] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil
pairing," in International Conference on the Theory and Application of
Cryptology and Information Security. Springer, 2001, pp. 514–532.

[19] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold

dss signatures," in International Conference on the Theory and Applications
of Cryptographic Techniques. Springer, 1996, pp. 354–371.
[20] A. Back et al., "Hashcash-a denial of service counter-measure," 2002.
[21] J. B. Kenney, "Dedicated short-range communications (dsrc) standards
in the united states," Proceedings of the IEEE, vol. 99, no. 7, pp. 1162–
1182, 2011.
[22] E. T. Lee and J. Wang, Statistical methods for survival data analysis.
John Wiley & Sons, 2003, vol. 476.
[23] A. Berkopec, "Hyperquick algorithm for discrete hypergeometric distribution,"
Journal of Discrete Algorithms, vol. 5, no. 2, pp. 341–347,
2007.
[24] J. A. Rice, Discrete Random Variables, ser. Mathematical Statistics
and Data Analysis. Cengage Learning, 2007, ch. 2.1, pp. 35–47,
2005938314. [Online]. Available: https://books.google.com/books?id=
KfkYAQAAIAAJ
[25] D. Zelterman, Models for discreet data. Oxford University Press, USA,
1999.