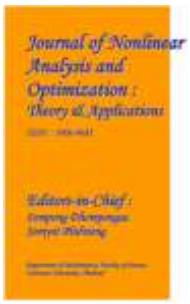
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 2 : 2024
ISSN : **1906-9685**



IDENTIFYING SYBIL ATTACKS VIA PROOFS OF WORK AND LOCATION MECHANISM IN VANETs

[1] SHAIK SADHIK [2] MR. B. SURESH

[1] PG Scholar in the department of MCA at QIS College of Engineering & Technology (AUTONOMOUS), Vengamukapalem, Ongole- 523272, Prakasam Dt., AP., India.

[2] Professor in the department of CSE/MCA at QIS College of Engineering & Technology (AUTONOMOUS), Vengamukapalem, Ongole- 523272, Prakasam Dt., AP., India.

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) are susceptible to various security threats, including Sybil attacks, which involve the creation of multiple fake identities by a malicious node to disrupt network operations. This paper proposes a novel approach for detecting Sybil attacks in VANETs by combining Proof of Work (PoW) and a location mechanism. The PoW scheme requires nodes to perform computationally intensive tasks to prove their identity, thereby making it difficult for adversaries to create multip`le fake identities. Additionally, the location mechanism utilizes the geographical position of vehicles to verify their authenticity, enhancing the detection accuracy. Through simulations and analysis, we demonstrate the effectiveness of our proposed approach in accurately identifying Sybil attacks while minimizing false positives and preserving network efficiency in VANET environments

INDEX ; vanet, pow, fake identities, accuracy

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

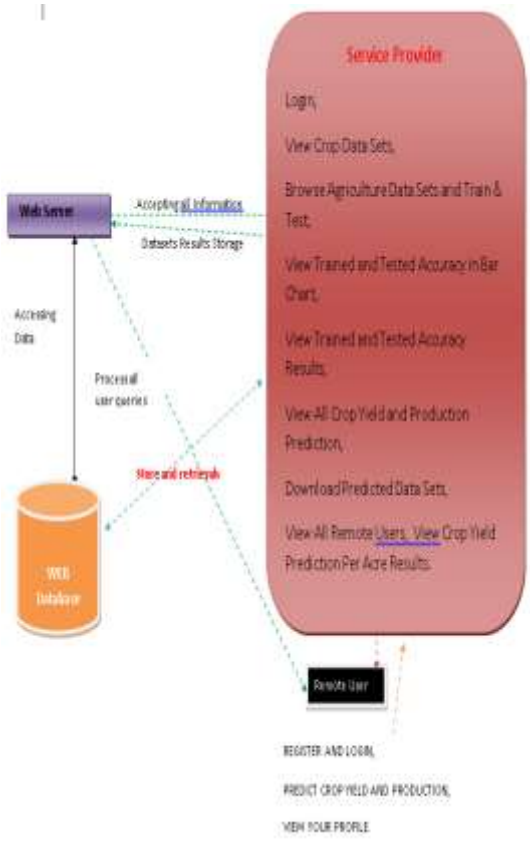
needs to compromise an infeasible number of RSUs to be able to create fake trajectories.

_ 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.

SYSTEM ARCHITECTURE



METHODOLGY

ALGORITHMS:

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

Logistic regression Classifiers

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.

Naïve Bayes

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.

Algorithm 1: Pseudo code for the random forest algorithm

To generate c classifiers:
for $i = 1$ to c **do**
  Randomly sample the training data D with replacement to produce D_i
  Create a root node, N_i containing D_i
  Call BuildTree(N_i)
end for

BuildTree(N):
if N contains instances of only one class **then**
  return
else
  Randomly select x% of the possible splitting features in N
  Select the feature F with the highest information gain to split on
  Create f child nodes of N, $N_1, \ldots, N_f$, where F has f possible values ($F_1, \ldots, F_f$)
  for $i = 1$ to f **do**
    Set the contents of N_i to D_i, where D_i is all instances in N that match F_i
    Call BuildTree(N_i)
  end for
end if

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.

Algorithm 1: SVM

1. Set $Input = (x_i, y_i)$, where $i = 1, 2, \cdots, N, x_i = R^n$ and $y_i = \{+1, -1\}$.
2. Assign $f(X) = \omega^T x_i + b = \sum_{i=1}^{N} \omega^T x_i + b = 0$
3. Minimize the QP problem as, $min\ \varphi(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + C.(\sum_{i=1}^{N} \xi_i)$.
4. Calculate the dual Lagrangian multipliers as $min\ L_P = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} x_i y_i(\omega x_i + b) + \sum_{i=1}^{N} x_i$.
5. Calculate the dual quadratic optimization (QP) problem as $max\ L_D = \sum_{i=1}^{N} x_i - \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j\ y_i\ y_j(x_i, x_j)$.
6. Solve dual optimization problem as $\sum_{i=1}^{N} y_i x_i = 0$.
7. Output the classifier as $f(X) = sgn(\sum_{i=1}^{N} x_i y_i(x \cdot x_i) +$.

RESULT ANALYSIS

Datasets Trained and Tested Results

Model Type	Accuracy
Naive Bayes	48.57142857142857
SVM	50.329670329670333
Logistic Regression	49.23076923076923
Decision Tree Classifier	43.956043956043956
Extra Tree Classifier	46.59340659340659

View Prediction Ratio Details

Bar chart prediction Result

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.

FUTURE ENHANCEMENT:

1.Improved Proof of Work (PoW): Enhance the computational puzzle in the PoW mechanism to make it more challenging for adversaries to generate multiple fake identities (Sybils). This could involve incorporating machine learning algorithms to dynamically adjust the difficulty level of the puzzle based on network conditions and traffic patterns.

2. Dynamic Location Verification: Integrate a dynamic location verification mechanism where vehicles periodically broadcast their GPS coordinates along with a cryptographic proof of location. This would make it harder for adversaries to spoof their location and launch Sybil attacks. Techniques such as secure multi-party computation or zero-knowledge

proofs can be employed to verify the authenticity of location information without revealing sensitive data.

3. Trust-Based Systems: Implement trust-based systems where vehicles build trust relationships based on past interactions and behavior. Vehicles can exchange reputation scores and use them to assess the trustworthiness of neighboring vehicles. Suspicious behavior such as rapid identity changes or inconsistent movement patterns can trigger further scrutiny**.**

4. Behavioral Analysis: Employ machine learning algorithms to analyze the behavioral patterns of vehicles in the network. Anomalies such as sudden changes in speed, route deviation, or irregular communication patterns can indicate the presence of Sybil attacks. Advanced anomaly detection techniques such as recurrent neural networks or deep learning models can be utilized for more accurate detection.

5. Blockchain Technology: Utilize blockchain technology to maintain a decentralized ledger of vehicle identities and transactions. Each vehicle can have a unique digital identity stored on the blockchain, and any attempt to create multiple identities (Sybils) would be easily detectable through consensus mechanisms. Additionally, smart contracts can be used to enforce identity management policies and penalize malicious behavior.

6. Collaborative Detection: Enable collaborative detection mechanisms where vehicles share information about detected Sybil nodes with each other. By leveraging the collective intelligence of the network, vehicles can quickly identify and isolate Sybil attacks before they cause significant harm.

7. Physical Layer Security: Explore the use of physical layer security mechanisms such as signal strength-based authentication or radio frequency fingerprinting to enhance the security of VANETs. These techniques rely on the unique characteristics of wireless communication channels to verify the authenticity of communicating nodes and detect Sybil attacks.

By integrating these advanced techniques and mechanisms, VANETs can significantly enhance their resilience against Sybil attacks and ensure the integrity and trustworthiness of the communication infrastructure.

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. Mahmou d, T. N. Guo, and M. Younis, "Cross-layerscheme 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 s rvice 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.

UTHOR PROFILE:

Mr. SHAIK SADHIK currently pursuing Master of Computer Applications at QIS College of engineering and Technology (Autonomous), Ongole, Andhra Pradesh. He Completed B.Sc. in Computer science from Acharya Nagarjuna University, Andhra Pradesh His area of interested is machine learning , cloud computing and DevOps

Mr.B.Suresh, currently working as an Associate Professor in the Department of Master of Computer Applications, QIS College of Engineering and Technology, Ongole, Andhra Pradesh. His area of interest is Machine Learning, Cloud Computing and Programming Languages