Rank and Location Based Security Mechanism for Mobile Networks

Gongjun Yan[1], Wu He[2], Hui Shi[3], and Dazhi Chong[4]
[1] Computer Science, University of Southern Indiana
[2] Information Technology Department, Old Dominion University
[3] Computer Information Systems, California State Polytechnic University
[4] MSIT, California Lutheran University
Email: gyan@usi.edu, whe@odu.edu, huishi@cpp.edu, dchong@callutheran.edu

***Abst*ract—In mobile networks, information security is of importance to most if not all applications. In this paper, we propose rank and geographic location based security mechanism in a mobile network scenario. Both sender and receiver are highly mobile and their public keys are known by all nodes. We extend the scheme of encryption/decryption on the basis of geo-encryption algorithm in [1]-[3] by adding information rank and person rank into consideration. The receiving nodes must be physically present in a designed decryption region and with correct security clearance rank to decrypt ciphertext within allowed time period. We also present algorithms to predict and update the decryption region in high mobility mode. The proposed method evaluated by simulations is efficient and secure.**

***Index Terms*—security, mobile networks, location security, encryption, decryption**

I. Introduction

Recent years have witnessed explosive increasing of applications in mobile networks. Most if not all applications needs sort of information security such as credit card, location, bank information, identity information, etc. In some extreme situations, we expect some encrypted messages can only be decrypted in a certain region, within certain time period, and by certain level security clearance rank people. The situations include but not limited to big transactions like house sale, legal communications, homeland defense incidents, even in war zone.

In the past there are some location-based algorithms such as GeoLock [4], [1], decryption region [3]. Many of these algorithms proposed effective algorithm for one or two parameters. In this paper, we proposed a security mechanism based on multiple factors including locations, time, information rank, and people rank. Since high mobility of nodes is one of unique feature in mobile networks, our proposed structure can be easily extended include mobility and other future factors. We also present decryption region can be predicted and updated among mobile nodes.

II. The State of Art

Location-based encryption method is proposed by Denning *et al.* [4], [1] that limits the area inside which the intended recipient can decrypt messages. Denning's geo-encryption model did not include details of an implementation of mobility support, so Al-Fuqaha *et al.* [5] proposed a model to provide for mobility when using GPS-based encryption. Yan *et al.* [3] also proposed location based encryption and decryption algorithm. In this algorithm, *GeoLock* is computed by the proposed algorithm which offers no-requirement on synchronizing key tables on all nodes in the network. Yan *et al.* [6], [2] proposed location security of nodes in vehicular networks by using Radar and history of record. PKI and digital signatures are well-explored methods in vehicular networks [7].

In this paper, we propose a security mechanism in which secret key is generated by both geographic location and security rank of visitors. Only when the visitor is physically present at the location region and with access security rank to be able to decrypt the ciphertext. Since nodes in mobile network sometimes move fast, we design the decryption region large enough to cover the error of the decryption region prediction. Moreover, we incorporate the prediction error by using location prediction deviation.

III. Rank and Location Based Security Mechanism

A An Overview

Our technique involves a security key handshake stage and a message exchange stage, as shown in Fig. 1. In the key handshake stage, the client and the server negotiate a shared symmetric key. The client generates two random numbers as keys *Key_S* and *Key_C*. *Key_S* is used to encrypt a message composed of the aggregated location message and *Key_C*. This encrypted message is *E*{*Req*}. The client generates a *GeoRankey* based on the location

Manuscript received January 7, 2019; revised May 26, 2019.

doi: 10.18178/ijsps.7.2.43-48

of the server. This value is XOR-ed with *Key_S* and then encrypted using the server's public key *Key_E* to produce the ciphertext *E*{*Key*}. Both *E*{*Req*} and *E*{*Key*} are transmitted to the server through the wireless channel. When the server receives *E*{*Key*}, it is decrypted using the server's private key *Key_D* to produce the XOR of the *GeoRankey* and *Key_S*. The *GeoRankey* generated from the GPS location of the server is used to recover the secret key *Key_S*. Then, *Key_S* is used to decrypt *E*{*Req*} to obtain the aggregated location message and the secret key *Key_C*.

In the message exchange stage, the server and client use the shared *Key_C* to communicate. When the server wants to reply to a client, it generates a random number, *Key_S*'. The reply message is directly encrypted using *Key_S*' to generate a ciphertext, *E*{*Rep*}. Since the aggregated location message contained the client's GPS position, the server can generate a *GeoRankey* of the client vehicle's decryption region. This *GeoRankey* is XOR-ed with *Key_S*' and then encrypted with *Key_C* to generate a ciphertext, *E*{*Key*'}. Both *E*{*Rep*} and *E*{*Key*'} are transmitted to the client through the wireless channel. *E*{*Key*'} is then decrypted using *Key_C* to recover the XOR of the client's *GeoRankey* region and *Key_S*'. The client generates its *GeoRankey* based on its current location. This is used to recover the secret key *Key_S*'. *E*{*Rep*} is decrypted using *Key_S*', and the reply message is recovered. The client repeats the algorithm in the message exchange stage to communicate with the server.

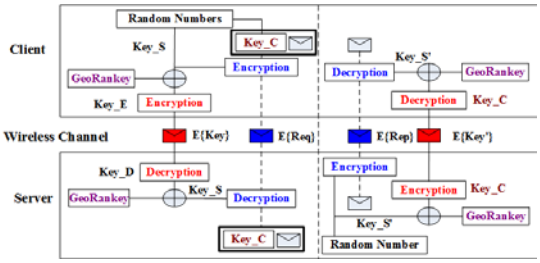

Figure 1. An overview of the proposed mechanism. *GeoRankey* makes extra security by enforcing both geographic location, information rank and person rank. Only the authorized people with correct rank at correct location and time are allowed to visit message.

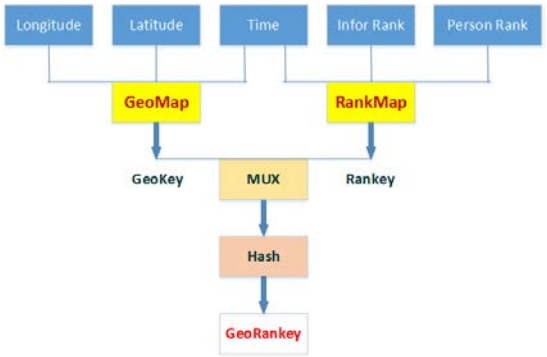

Figure 2. *GeoRankey* generation structure. The receipient's geographic location, time, information security rank and person rank are computed together to generate a unique key, *GeoRankey*.

GeoRankey is generated by multiple pieces of information such as geographic location, time, information security rank and person rank. The most important function in the proposed mechanism is to map geographic location and time into a unique key, i.e. *GeoKey* and to map information rank, person rank, and time into a unique key, i.e. *Rankey*. The two keys *GeoKey* and *Rankey* are multiplexer-ed (MUX-ed) together and then hashed into a final secret key, i.e. *GeoRankey*, as shown in Fig. 2. In this key, any information missed will result a different *GeoRankey* which further enhance security.

IV. Mapping Information to Secret Key

There are two ways to map information to secret keys. One way is to use several encrypted key tables which are synchronized among all nodes to locate unique secret keys. Another way is to use an algorithm to generate unique secret keys.

A Key Table

Sharing secret keys by using key tables is often effective and straightforward. If all nodes in the system are equipped with exactly same key tables as shown in Fig. 3 at any time, we can ensure the secret keys can be obtained by checking the table and inputing longitude and latitude at certain time [1].

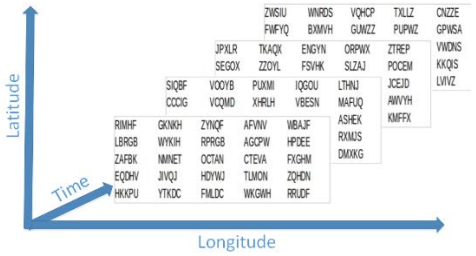

Figure 3. Geographic Location Table. We refer geographic longitude and latitude at certain time to obtain a unique secret key.

Similarly, we can refer information rank and people's security clearance rank at certain time to a rank table shown as in Fig. 4. This rank table defines that a person with a certain security clearance rank can access information with a certain rank. Notice that higher rank value does not override lower rank value in this table. Each rank secret key is checked by exact information rank and person rank.

Figure 4. Rank Table. We refer information rank and people's security clearance rank at certain time to obtain a unique secret key.

An example of applying the key table is shown in Fig. 5. The decryption region is along the waypoints of a node. Therefore, we need to check both location table and rank table to obtain the secret keys for each waypoint.

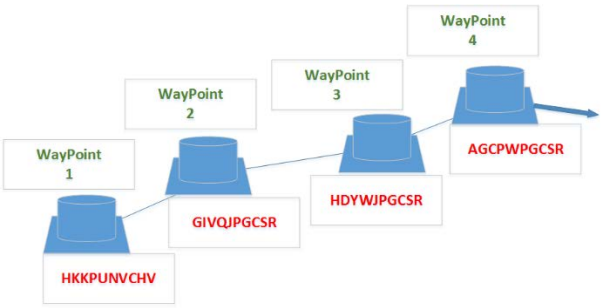

Figure 5. Secret keys are obtained by checking both location table during a specific time period. The *GeoKey* and *Rankey* are combined to serve as a secret key for waypoints.

One challenge of Key tables is to synchronize all tables of all nodes where there are updates in mobile networks. Failed to synchronize updates among all nodes will result failure of encryption and decryption.

B Key Algorithm

Key algorithm generate secret keys by taking input parameters. It has an advantage to overcome synchronization among all nodes. The algorithm can be firm-wired into hardware and works like a black-box to nodes. Fig. 6 showed a simple algorithm that can compute a unique key from location, velocity, and time information.

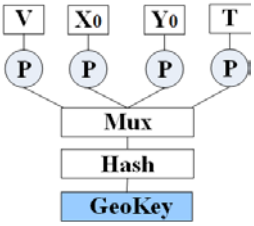

Figure 6. Generate a unique key from location, velocity, and time. It can define a range of location and time and a certain time period.

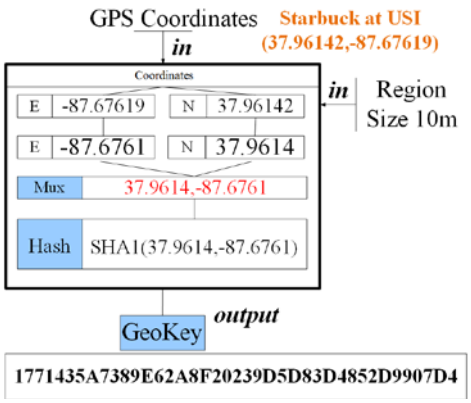

Figure 7. Example to convert (37.96142,−87.67619) into a unique *GeoKey* which specifies a square with size of 10 by hash SHA1.

More specifically, we use an example of Starbuck at 8600 University Blvd, Evansville, IN 47712, USA is (37.96142,−87.67619) in decimal degrees format, i.e. Latitude: 37.96142 Longitude: −87.67619 to compute a secret key from location information. In this particular example, we want to convert (37.96142,-87.67619) into a unique *GeoKey* which specifies a square with size of 100. Since degree precision versus length for 0.0001 is about a parcel, so our region is about 10 meter square.

On the other hand, it is challenging to design sophisticated key algorithm that can be resistant to reverse-engineering. This will remain as future work.

V. Decryption Region in Vehicular Networks

The geo-encryption protocol allows nodes to securely communicate with nodes at a particular location and time period. We enhance the geo-encryption methods by the special features of vehicular networks. In this paper, we have two improvements of determining decryption region: predicting and updating decryption region. The movement of nodes is constrained by roads, and the map of the roads can be accessed by all nodes. Therefore, we can predict nodes' position based on the map and nodes' mobility. Based on the prediction of decryption region, the communication messages are checked by geographic location. Because of dynamics of nodes, there will be a certain prediction error. Therefore the predicted decryption region will be corrected by updating the real positions. The real positions are piggy-backed by communication messages.

A Prediction of the Decryption Region

Suppose the target decryption region starts from position $P_0(x_0,y_0)$. The decryption region is assumed as a square region. Since a square region must have two components: starting point, length (length equals to width). Since the starting position can be predicted by checking maps and mobility of nodes, only the length of square needs to be determined. The length of square is listed as a series of scales: L, for example, 10, 20,..., 1000 meters. For 10 digits UTM positions, $1<L<10^4$ because the precision is about 1 meter. For 8 digits UTM positions, $10<L<10^7$ because 8 digits UTM positions are accurate to 10 meters ([8]). For 6 digits UTM positions, $L<10^4$ because 6 digits UTM positions are accurate to 100 meters. No smaller than 6 digits can be use in our proposal. Therefore, the length of square is selected from one of the three possible lengths: 10, 100, 1000 meters.

The decryption region can be predicted in several ways in vehicular networks based on the map of roads and mobility of nodes. The methods that predict the receiver's region are the following.

1) The location of communication peers can be calculated on the basis of the mobility parameters including current speed, current position, current acceleration, etc. This is a major method. A new position after a certain time interval can be computed. Suppose at time t_0, the target vehicle is at location (x_0,y_0) with speed v_{x0},v_{y0} and acceleration

a_{x0},a_{y0}, where x_0,v_{x0},a_{x0} are the x-axis value of initial location, relative speed on x-axis direction, and relative acceleration on x-axis direction; y_0,v_{y0},a_{y0} are the y-axis value of initial location, relative speed on y-axis direction, and relative acceleration on y-axis direction;. After time interval *t*, we can roughly predict that the vehicle will at a place near location region: x_1, or

$$x_1 \in [x_0 + v_0 t + \frac{1}{2} a_0 t^2 - \alpha * XDeviation, \\ x_0 + v_0 t + \frac{1}{2} a_0 t^2 + \alpha * XDeviation] \quad (1)$$

and y_1, or

$$y_1 \in [y_0 + v_0 t + \frac{1}{2} a_0 t^2 - \alpha * YDeviation, \\ x_0 + v_0 t + \frac{1}{2} a_0 t^2 + \alpha * YDeviation] \quad (2)$$

where x_1,*XDeviation* are the location prediction on x-axis and the deviation of position value of x-axis; y_1,*YDeviation* are the location prediction on y-axis and the deviation of position value of y-axis and α is the coefficient which implies the affection of the deviation, 0≥α≥1.

2) If the decryption region is a fixed area, we can directly check the map of roads and calculate the GPS coordinates. This is the simplest scenario. Usually the e-business location is known on digital map. A location of a new business can be registered by the digital map generator.

3) If the decryption region is dynamically moving, we can calculate the position of the decryption region by querying the target receiver. This method is addressed in ([5], [9]).

B Updating The Decryption Region

Although the decryption region is predicted, there are prediction errors of decryption region because of dynamics of nodes. Therefore, the decryption region needs to be corrected to improve prediction precision for next communication. The predicted position will be updated by the real position which is piggybacked in communication messages. The speed, acceleration and direction of move will be piggybacked as well. Therefore, the updating step includes the following assignment:

$$x_1 = x_{real} \quad (3)$$

$$y_1 = y_{real} \quad (4)$$

$$XDeviation = (1-\beta) * XDeviation + \beta * |x_{real} - x_0| \quad (5)$$

$$YDeviation = (1-\beta) * YDeviation + \beta * |y_{real} - y_0| \quad (6)$$

where (x_{real},y_{real}) is the real position piggybacked, β is the coefficient value which implies the effect of the prediction error $|x_{real} - x_0|$.

The updating frequency is depended on the mobility of receiving nodes, the precision requirement of decryption region and the bandwidth of control channel. For example, the frequency of updating on highways is much higher than the frequency of updating on urban area because the velocities on highway are much higher than the ones in urban area. Similarly, precision of decryption region and the bandwidth of control channel impact the updating frequency as well.

VI. Simulation Settings

We used SUMO [10] and ns-2 [11] for our simulation. The simulator SUMO is an open source and microscopic road traffic simulation package. The simulator ns-2 is a well-known open source and the second version of network simulator. The SUMO creates a trace file which records the mobility of nodes. We loaded the trace file into ns-2 to simulate the security. The application is Constant Bit Rate (CBR) with 16 packets every second. The total amount of nodes is 320. The map is 3.2km x 3.2km. The decryption region of roadside shop is a square of 10m x 10m. Vehicle's decryption region is determined by roadside shops and dynamically changed on the basis of vehicle's mobility. The size of vehicular decryption region is 3m x 3m. The square decryption region is initially set in the middle of the simulation area, shown as Fig. 8. We assume that nodes can decrypt a message in the region.

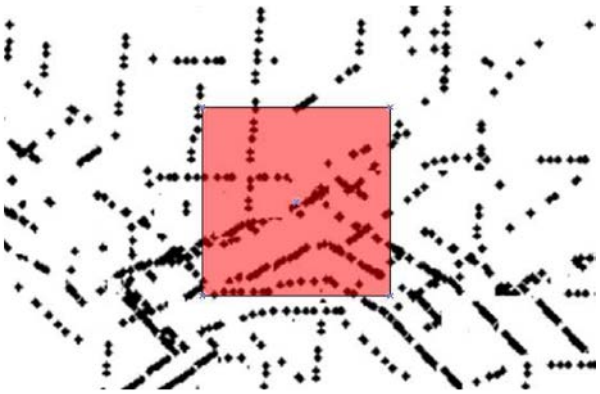

Figure 8. Decryption region snapshot (not proportionally drawn).

First, we expect to investigate the decryption ratio over location precision. We calculated the decryption ratio by counting failed decrypted messages over total ciphertext received. This ratio is not the delivery ratio but the decryption ratio inside the decryption region. We intentionally varied the location precision because location has precision issues in GPS. The square size is set as 10 meters for roadside shops and 3 meters for nodes. We compared two set of results with different speed (24 meters per second and 14 meters per second) which is shown in Fig. 9. As we expected, the increase of location tolerance will cause the decrease of the decryption rate. Besides, the faster speed will cause lower decryption rate. This is because that the increase of location tolerance and the increase of speed will cause the increase of false location of nodes. We noticed that

Geokey from Geographic location algorithm (GeoAlgKey) as shown in Fig. 7 has lowest decryption errors because the algorithm has certain precision tolerance function.

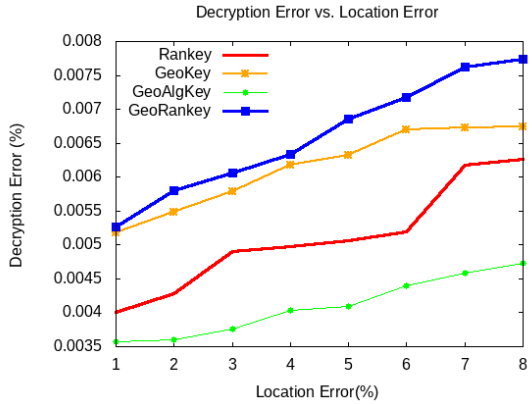

Figure 9. Decryption ratio.

Security is normally at certain cost. We measured time spent to decrypt messages that is encrypted by 3DES and keys generated in this paper, such as Rankey, Geokey, Geographic location algorithm (GeoAlgKey), and GeoRankey. We changed the size of message from 25KB to 2MB, increment 25KB each time. We measured decryption time overhead after a ciphertext message is received. Therefore, decryption time includes both the secret key recovery, validation, and actual decryption. Fig. 10 shows the result that time overhead increases quickly when the sizes of messages increase. It turns out that the algorithm GeoAlgKey presented in Fig. 7 has lowest time consumption. The GeoRankey has largest time overhead which makes sense because of two table search, when the message sizes increase. Overall, time overhead for decryption is in normal range and is acceptable.

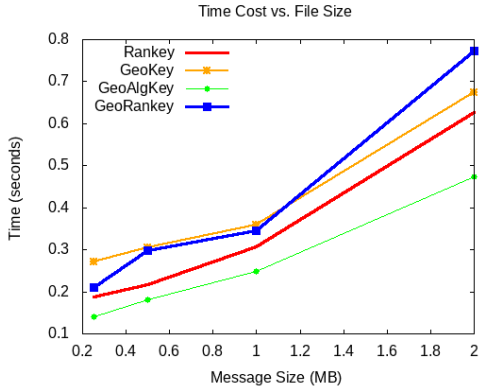

Figure 10. Decryption time.

VII. Conclusion

In this paper, we present a novel security mechanism in which multiple factors including security clearance rank, information rank, geographic location and time are compted to generate a secret key. This paper is further extended concepts proposed by in our previous work [2], [3]. The future work will further explore general mapping function to generate secret key from other factors such as mobility, road information etc.

References

[1] L. Scott and D. E. Denning, "Location based encryption technique and some of its applications," in *Proc. Institute of Navigation National Technical Meeting 2003*, Anaheim, CA, 2003, pp. 734-740.

[2] G. Yan, D. Wen, S. Olariu, and M. C. Weigle, "Security challenges in vehicular cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 284-294, 2013.

[3] X. Liu, W. He, L. Xu, and G. Yan, "Enhancing the security of cloud manufacturing by restricting resource access," *Journal of Homeland Security and Emergency Management*, vol. 11, no. 4, pp. 533-554, 2014.

[4] D. Denning and P. MacDoran, "Location-based authentication: Grounding cyberspace for better security," *Computer Fraud and Security*, vol. 1996, no. 2, pp. 12-16, 1996.

[5] A. Al-Fuqaha and O. Al-Ibrahim, "Geo-encryption protocol for mobile networks," *Comput. Commun.*, vol. 30, no. 11-12, pp. 2510-2517, 2007.

[6] G. Yan, S. Olariu, and M. C. Weigle, "Providing location security in vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 16, no. 6, 2009.

[7] D. Qiu, S. Lo, P. Enge, and D. Boneh, "Geoencryption using loran," in *Proc. ION NTM 2007*, January 22-24, 2007.

[8] J. N. G. Terry, "How to read the universal transverse mercator (utm) grid," *GPS World*, April 1996, p. 32.

[9] L. Scott and D. Denning, "Geo-encryption: Using gps to enhance data security," *GPS World*, April 1 2003.

[10] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. (December 2012). Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal on Advances in Systems and Measurements*. [Online]. 5(3-4). pp. 128-138. Available: http://sumo.sourceforge.net/

[11] The Network Simulator NS-2. [Online]. Available: http://www.isi.edu/nsnam/ns/

Gongjun Yan received his Ph.D. in Computer Science from Old Dominion University in 2010. He is currently an Assistant Professor in University of Southern Indiana and has been working on the issues about Intelligent Transportation, Vehicular Ad-Hoc Networks, Sensor Networks, Wireless Communication, and Machine Learning. His main research areas include intelligent vehicles, security, privacy, routing, and intelligent systems. In years, he applies mathematical analysis to model behavior of complex systems and integrates existing techniques to provide comprehensive solutions. He had more than 60 publications including journal/conference papers, book chapters, and patents. In recent three years, he has won three best paper awards in international conferences. He also serves as associate editors in journals such as IEEE Transaction on Intelligent Transportation System, Ad Hoc & Sensor Wireless Networks, etc.

Wu He received the B.S. degree in Computer Science from DongHua University, China, in 1998, and the Ph.D. degree in Information Science from the University of Missouri, USA, in 2006. He is an Associate Professor of Information Technology at Old Dominion University, Norfolk, VA, USA. His research interests include Data Mining, Information Security, Social Media, Knowledge Management and Computing Education. His research has been funded by NSF, NSA, NASA and other organizations. He has published articles in prestigious journals such as Information & Management, Journal of the Association for Information Science and Technology, International Journal of Information Management, and IEEE Transactions on Industry Informatics.

Hui Shi received the B.S. degree in Computer Science and Technology from Hefei University of Technology, China, in 2003, the M.S. degree in Computer Application Technique from Hefei University of Technology, China, in 2006, and Ph.D. degree in Computer Science from Old Dominion University, USA, in 2014. She is an Assistant Professor of Computer Information Systems at California State Polytechnic University, Pomona, CA, USA. Her research interests include Data Mining, Social Media Data Analytics, Semantic Web, Big Data, Computer Supported Cooperative Work, and Information Visualization. Her research has been funded by NSF and other organizations. She has published articles in prestigious computer science and information systems journals and international conference proceedings.

Dazhi Chong received Ph.D in information technology from Old Dominion University, Norfolk, Va, M.S. in computer application technique from Hefei University of Technology, China. Currently he is an assistant professor of the Master of Science in Information Technology program at California Lutheran University. Before joining CLU, he was an adjunct faculty in IT department of Old Dominion University. Dazhi Chong's research covers a wide range of topics in the IT discipline, including Information Systems, Social Network Analysis, Computer Supported Cooperative Work, Data Mining, and Financial Analysis. He has published papers in a wide range of scholarly refereed journals and conference.