International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1553

VOLUME 3, ISSUE 5

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

Editor in Chief Dr. Haralambos Mouratidis

International Journal of Computer Science and Security (IJCSS)

Book: 2009 Volume 3, Issue 5
Publishing Date: 30-11-2009
Proceedings
ISSN (Online): 1985-1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 3, Issue 5, November 2009.

Pages

Performance Analysis of Mobile Ad-hoc Network Using AODV Protocol

Dr. Aditya Goel adityagoel2@rediffmail.com
Department of Electronics & Communication Engineering
Maulana Azad National Institute of Technology
(Deemed University)
Bhopal, India - 462051

Ajaii Sharma ajaiisharma@yahoo.com
Department of Information Technology
Maulana Azad National Institute of Technology
(Deemed University)
Bhopal, India - 462051

Abstract

This research work proposes a new protocol that modifies AODV to improve its Performance using *Ant Colony algorithm*. The mobility behaviour of nodes in the application is modelled by the random waypoint model through which random locations to which a node move are generated, and the associated speed and pause time are specified to control the frequency at which the network topology is changed. The *Optimized-AODV* protocol, incorporates path accumulation during the route discovery process in AODV to attain extra routing information. It is evident from the results that *Optimized-AODV* improves the performance of AODV under conditions of high load and moderate to high mobility.

Keywords: MANET, Optimized-AODV, Ant Colony algorithm (ACO),

1.0 Introduction

Today's Internet has been developed for more than forty years. Recently many researchers are studying networks based on new communication techniques, especially wireless communications. Wireless networks allow hosts to roam without the constraints of wired connections. People can deploy a wireless network easily and quickly. Hosts and routers in a wireless network can move around. Wireless networks play an important role in both military and civilian systems. In the recent years Mobile Ad- hoc network has found applications especially to overcome the limitation of Bandwidth in wireless communication.

One of the main difficulties in MANET (Mobile Ad hoc Network) is the routing problem, which is aggravated by frequent topology changes due to node movement, radio interference and network partitions. Many Routing protocols have been proposed in past and reported in the literature [1]. The proactive approaches attempts to maintain routing information for each node in the network at all times [2, 3], where as the reactive approaches only find new routes when required [5, 6, 7, 8] and other approaches make use of geographical location information for routing [8]. The biological swarms like ants or honeybees often contain thousands of individuals [10, 11, 12]. They perform extraordinarily complex tasks of global optimization and resource allocation using only local information. The wireless network topology is dynamic and unpredictable. Traditional routing protocols used for wired networks cannot be directly applied to most wireless networks[22] because some common assumptions are not valid in this kind of dynamic network, like a node can receive any broadcast message sent by others in the same subnet. The bandwidth in this kind of network is usually limited. Although the researchers have suggested other techniques to enhance the overall network bandwidth by integrating wireless network with Optical network [23]. Thus, this network model introduces great challenges for routing protocols. There are some algorithms that use ant-like mobiles agents to maintain routing and topology discovery for both wired and wireless networks [13]. We focus on improving performance of the reactive ad hoc routing protocols using the ideas from swarm intelligence, particularly the ant colony Meta heuristic.

2.0 AODV System

AODV (Ad-hoc On-demand Distance Vector) is a loop-free routing protocol for ad-hoc networks. It is designed to be self-starting in an environment of mobile nodes, withstanding a variety of network behaviours such as node mobility, link failures and packet losses. The AODV protocol consists of two important mechanisms, Route Discovery and Route Maintenance. AODV is chosen for the obvious reason that it is simple and has a low overhead and its on-demand nature does not unduly burden the networks.

The optimized Ant Colony protocol has been designed using communication based design methodology. The first step in this design flow is the capture of specifications and functional decomposition at the system level.

The optimization of AODV is based on the recent draft of the AODV specification [4]. The essential functionality of AODV includes:

- *RREQ* and *RREP* messages (for route discovery)
- *RERR* messages, HELLO messages, & precursor lists (for route maintenance)
- Sequence numbers
- Hop counts
- Expanding ring search

The following fields exist in each route table entry of AODV:

- *Destination IP Address*: The IP address of the destination for which a route is supplied
- *Destination Sequence Number*: It is associated to the route.
- *Next Hop*: Either the destination itself or an intermediate node designated to forward packets to the destination
- *Hop Count*: The number of hops from the Originator IP Address to the Destination IP Address
- *Lifetime*: The time in milliseconds for which nodes receiving the RREP consider the route to be valid
- *Routing Flags*: The state of the route; up (valid), down (not valid) or in repair.

Suppose S would like to communicate with D Figure 1, the node broadcasts a RREQ to find a route to the destination. S generates a Route Request with destination address, Sequence number and Broadcast ID and sent it to his neighbour nodes. Each node receiving the route request sends a route back (Forward Path) to the node.

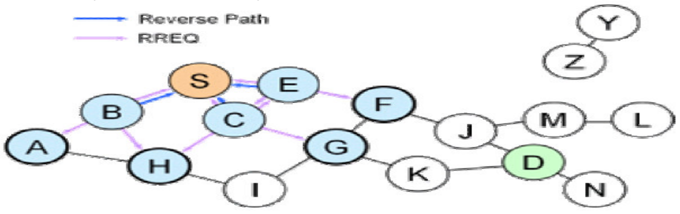


FIGURE 1: Path finding in AODV

A route can be determined when the RREQ reaches a node that offers accessibility to the destination, e.g., the destination itself.

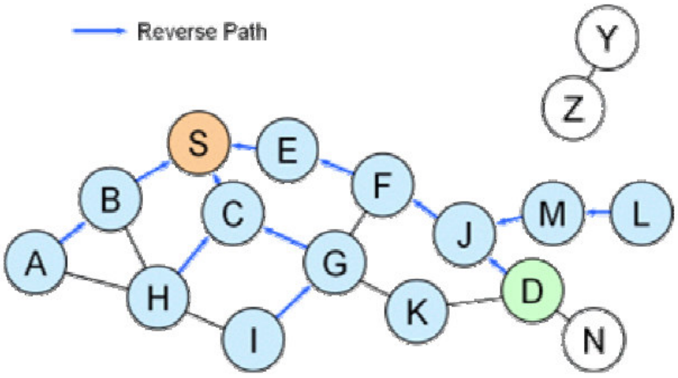


FIGURE 2: Path finding in AODV

The route is made available by unicasting a RREP back to D and is written in the routing table from S Figure 2. After receiving the route reply every node has to update its routing table if the sequence number is more recent.

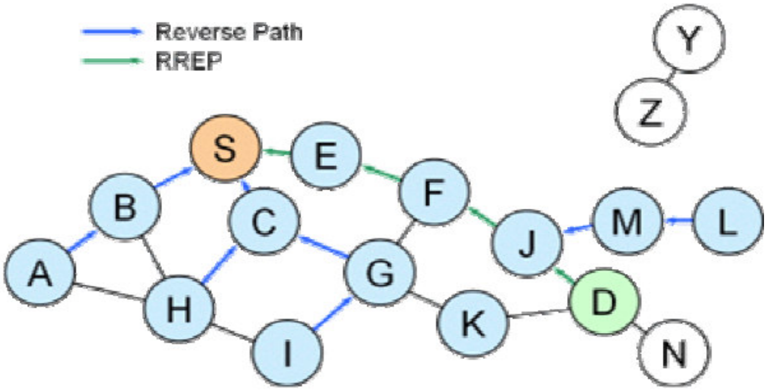


FIGURE 3: Path finding in AODV

Now node S can communicate with node D, Figure 3, 4.

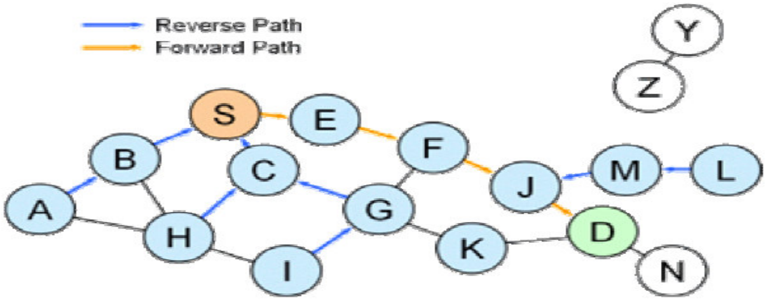


FIGURE 4: Path finding in AODV

When a link break in an active route is detected, the broken link is invalid and a RERR message is sent to other nodes, Figure 5. If the nodes have a route in their routing table with this link, the route will be erased. Node S sends once again a route request to his neighbour nodes. Or a node on the way to the destination can try to find a route to D. That mechanism is called: *Local Route Repair.*

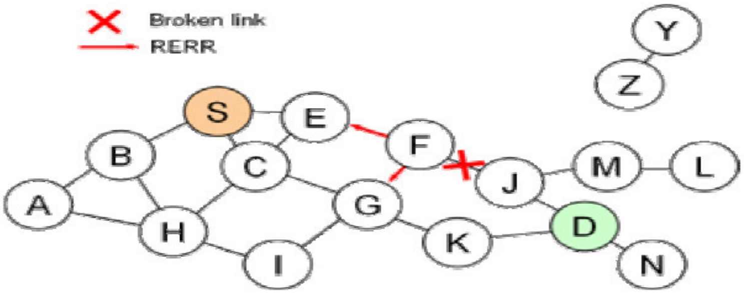


FIGURE 5: Path finding in AODV

The *Ant Colony Algorithm (ACO)* [10, 13] is designed to use the Object oriented tool command language. The following set of core properties characterizes ACO instances for routing problems:

- Provide traffic-adaptive and multi path routing,
- Rely on both passive and active information monitoring and gathering,
- Make use of stochastic components,
- Do not allow local estimates to have global impact,
- Set up paths in a less selfish way than in pure shortest path schemes favouring load balancing,
- Show limited sensitivity to parameter settings.

2.1 Optimized AODV

In the optimized protocol, the interactions of ant like packets are used to proactively maintain the un-expired route connectivity following the stigmergy paradigm. The artificial ants (ant-like packets) are divided into two classes: *forward ant* and *backward ant*. The *forward ants* traverse the network to collect network traffic information, which mimics the ant in the searching mode and the *backward ants* utilize this information to update routing tables and other data structures, which mimics the ant in the carry mode. For simplicity, it is assumed that all of the *forward ants* will eventually find the destination and do not consider the ants in return mode. At the same time as using the proactive process to maintain the unexpired route connectivity, the reactive features of the original AODV protocol are retained for the new route discovery and route error handling.

Simulation Parameters

- IEEE 802.11 is used as the MAC layer protocol.
- The radio model simulates with a nominal bit rate of 2Mbps
- Nominal transmission range of 250 meters.
- The radio propagation model is the two-ray ground model.
- First 100 nodes are deployed for one experiment and then 1000 nodes are used for another experiment in a rectangular field of 3000m X 600m.
- The traffic pattern is CBR (constant bit rate) with a sending rate of 4 packet/seconds and the packet lengths are all 512 bytes.
- The mobility model used is the *Random Waypoint Model*
- Each node moves with a randomly chosen speed uniformly from 1-20 m/sec.
- The *pause time* of the node mobility as the independent variable that reflects the degree of the node mobility. The small pause time means intense node mobility and large pause time means slow node mobility. The ant trip time *T is used* as the reinforcement signal
- The simulations are performed with 7 different pause times 0, 30, 60, 120, 300 and 600 seconds.
- The simulation time is 600 second.

To avoid the broadcast storm problem, the gossip-based probability scheme of [14] is used. The Network Simulator 2 [15, 18] is used as the simulation tool.[17]

3.0 Results and Discussion

The investigation has been carried out by mainly considering the following four different scenarios:
1. The packet delivery ratio
2. The average end-to end delay
3. The normalized routing overhead
4. Throughput

3.1 The packet delivery ratio

With the first scenario the robustness of the routing protocol are compared

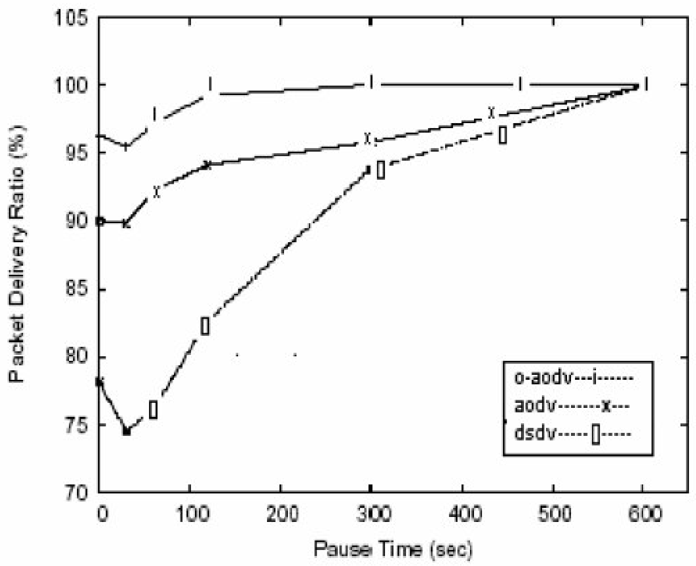


FIGURE 6: Packet Delivery Ratio with 100 nodes

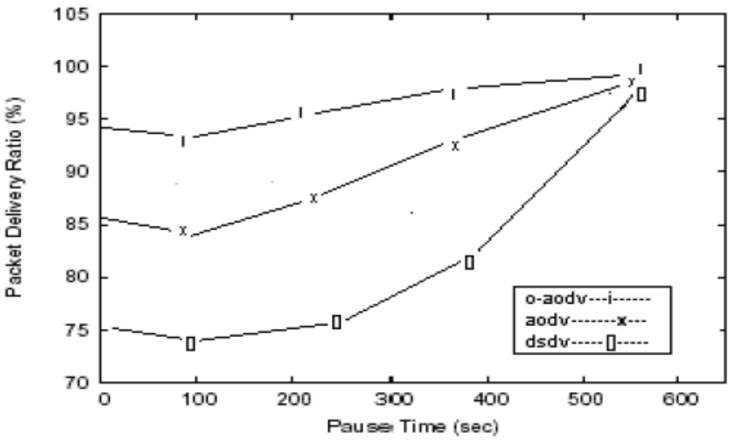


FIGURE 7: Packet Delivery Ratio with 1000 nodes

Fig. 6 and Fig. 7 show the delivery ratio comparisons, for *Optimized-AODV*, AODV and DSDV for 100 nodes and 1000 nodes respectively. In the case of high mobility, only *Optimized-AODV* performs well, delivering most of the data packets. DSDV performs worst due to its simply dropping data packets for which there is no valid route. With the mobility decreasing, the routing status becomes relatively stable. Fig. 7 shows the case of higher node density, in this case performance of DSDV is worst, *and Optimized-AODV* still gives stable performance, as more numbers of paths are available to send the packets. At the low mobility end the performance of all three protocols is close. When nodes are not moving path finding process is not required. Hence we observe that *Optimized-AODV* gives stable results.

3.2 The average end-to end delay

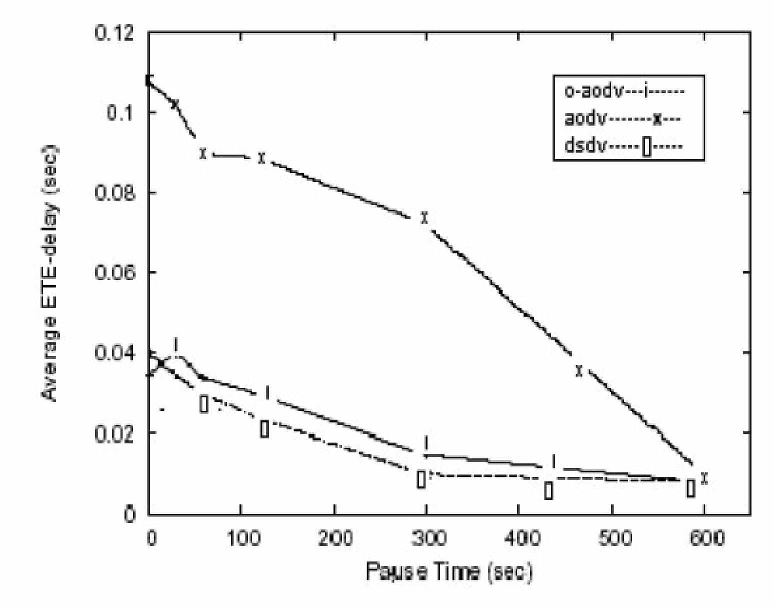


FIGURE 8: Average ETE-delay with 100 nodes

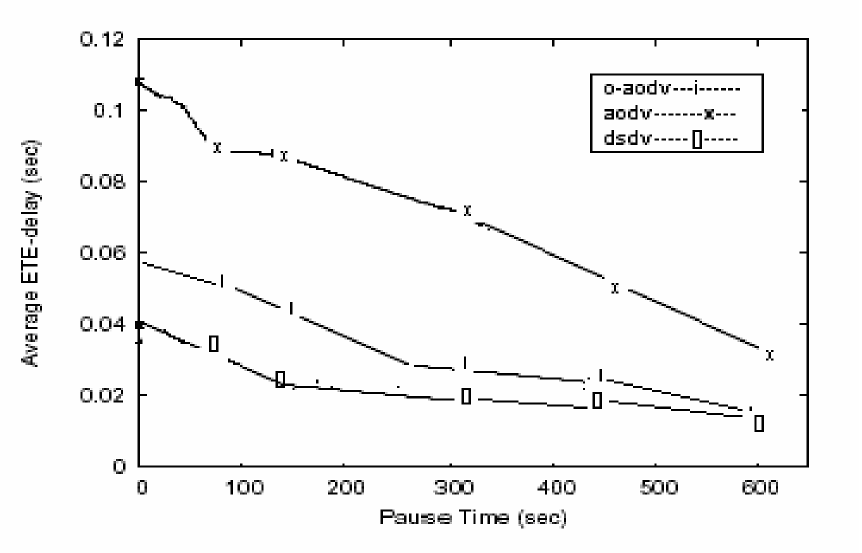


FIGURE 9: Average ETE-delay with 1000 nodes

The quality of service provided by the protocols can be compared with the average delay. Fig.8 and Fig.9 present the comparisons of the average end-to-end delay for *Optimized-AODV*, AODV and DSDV for 100 and 1000 nodes. The average ETE delay decreases with reduced mobility for all three protocols. AODV shows largest delays in situations with high mobility due to its single path nature and inefficient manner to handle route failure. *Optimized-AODV*, on the other hand, shows low delays in all cases. This is because, instead of buffering data packets for a new route to be found, *Optimized-AODV* forwards the data packets through alternative routes. DSDV exhibits a low delay because of its continuous maintenance of routing information and no data buffering for the data without valid routes. With the higher node density, overall end-to-end delay for all the cases increases as number of hopes increases. In these cases the packet needs to cover long distance to reach the destination.

3.3 The normalized routing overhead

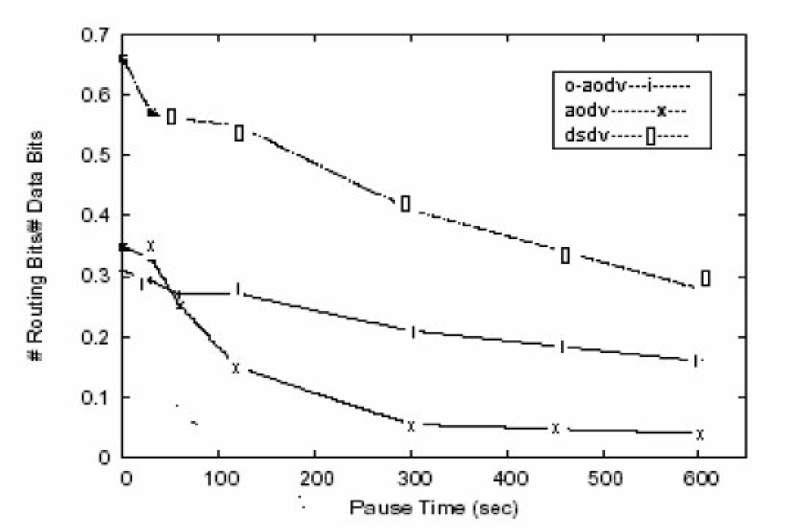


FIGURE 10: Routing Over head with 100 nodes

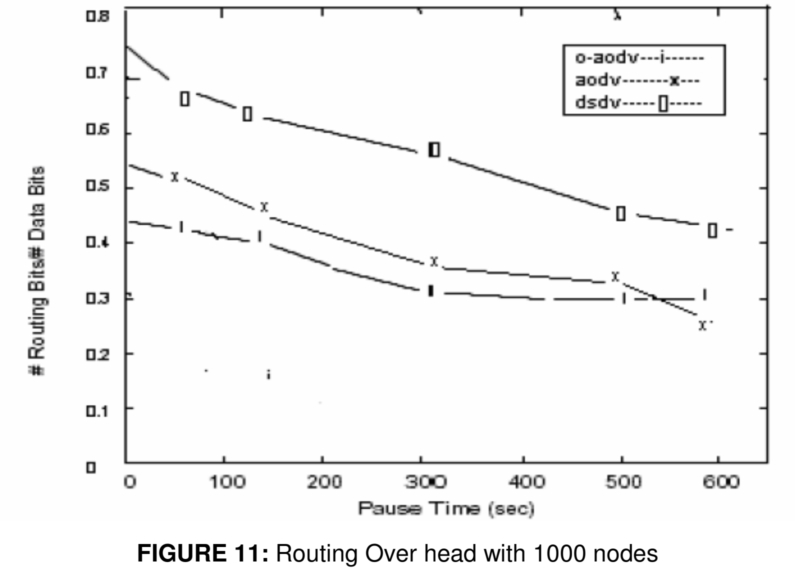


FIGURE 11: Routing Over head with 1000 nodes

Fig. 10 depicts the number of control packets per data packet needed to perform the routing work for *Optimized-AODV*, AODV and DSDV. Bits used for routing, is counted because the different routing protocols generate the routing overhead in very different ways. In the case of very high mobility it is obvious that *Optimized-AODV* creates the least overhead compared with AODV and DSDV. With high node mobility, route failure occurs more frequently, and AODV will cause flooding of large number of route finding packets, while the number of routing packets in *Optimized-AODV* is independent of node mobility. With less mobility, the performance of *Optimized-AODV* still remains stable and the overhead of AODV is slightly less than *Optimized-AODV*. DSDV shows a very high difference in comparison to the other two protocols. In the case of high-density as shown in Fig. 11 over all routing over heads increases. Here *Optimized-AODV* is more advantageous as it gives minimum overheads.

3.4 Throughput

Throughput comparison between three protocols with 300 pause time

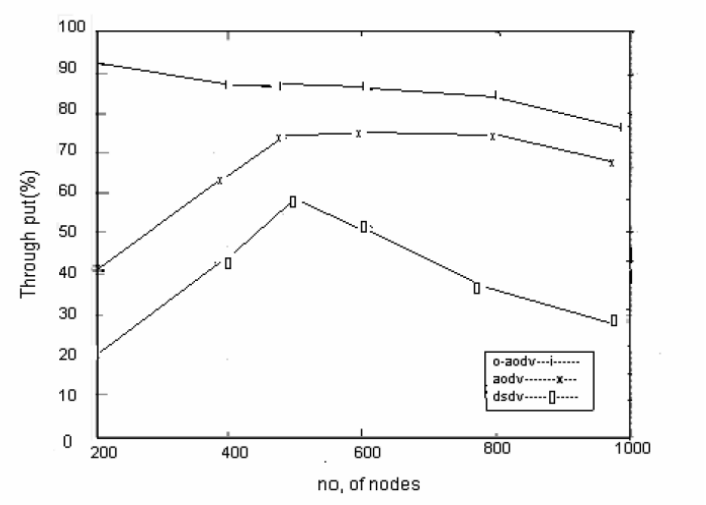


FIGURE 12: Throughput comparison

Fig. 12 shows the throughput comparison of three different protocols. *Optimized–AODV* shows approximately constant graph indicating the scalability of protocol. With the change in density its throughput is stable. In case of AODV protocol when number of nodes increases, initially throughput increases as large number of routes are available but after a certain limit throughput becomes stable due to increase in end-to-end delay. DSDV gives comparatively lower throughput as the large number of routing bits is required. Increase in overhead reduces the throughput.

The above investigation and subsequent discussion reveals that *optimized-AODV* is capable of overcoming the limitations posses by AODV and DSDV algorithms. Our study therefore recommends optimized-AODV protocol for routing of data at different dynamic nodes of large mobile ad hoc network.

4.0 Conclusion and Future scope

During the study of packet delivery ratio of data packets, *Optimized-AODV* scales better than AODV in large networks. The performance of *Optimized-AODV* remains stable, for low node density as well as in the high node density. At the low mobility end the performance of all three protocols is close. When nodes are not moving path finding process is not required. During the study of End to End delay, *Optimized-AODV* shows low delays in all cases, as instead of buffering data packets for a new route to be found, *Optimized-AODV* forwards the data packets through alternative routes. During the study of routing overhead, it was found that with high node mobility route failure occurs more frequently, and AODV will cause flooding of large number of route finding packets, while the number of routing packets in *Optimized-AODV* is independent of node mobility. With less mobility, the performance of *Optimized-AODV* still remains stable and the overhead of AODV is slightly less than *Optimized-AODV*. DSDV shows a very high difference in comparison to the other two protocols. In the throughput comparison, *Optimized–AODV* shows approximately constant graph, which indicates the scalability of *Optimized-AODV* protocol. With the change in density its throughput is stable. In case of AODV protocol when number of nodes increases, initially throughput increases as large number of routes are available, after a certain limit throughput becomes stable due to increase in end-to-end delay. DSDV gives comparatively lower throughput as the large number of routing bits is required. Increase in overhead reduces the throughput. The difference in the routing load of *Optimized-AODV* and DSDV decreases with an increase in the load. *Optimized-AODV* can be used either as an alternative to AODV or as an optimization under moderate to high load scenarios. Based on these simulated results, it is clear that the *Optimized-AODV* could also be suitable if overall routing load or if the application oriented metrics such as delay and packet delivery ratio are important consideration for the ad hoc network application. Optimized AODV is recommended as a better protocol especially for large Mobile Ad hoc Networks.

This protocol can be tested for real data set. We have discussed the behaviour of our proposed ant algorithm with mobility model that represent multiple mobile nodes MANETs whose actions are completely independent of each other. One of our future research studies is the study of the behaviour of our proposed algorithm with mobility models such as Point Group Mobility model which represents multiple MANETs moving together [20] and similar swarm based clustering [19,21].

References

[1] R.Asokan, A.M.Natragan, C.Venketesh, “Ant Based Dynamic Source Routing Protocol to Support Multiple Quality of Service (QoS) Metrics in Mobile Ad Hoc Networks” International Journal of Computer Science and Security, volume (2) issue (3)

[2] C. E. Perkins and P. Bhagwat, “Highly dynamic destination sequenced distance-vector routing (dsdv) for mobile computers” ACM SIGCOMM: Computer Communications Review, vol. 24, no. 4, pp. 234–244, 1994.

[3] P.Jacquet, P.Muhlethaler, and A.Qayyum, “Optimised link state routing protocol” IETF MANET, Internet Draft, 1998.

[4] C.Perkins, E.Belding-Royer, S.Das “Ad hoc On-Demand Distance Vector (AODV) Routing” Feb.2003.http://www.ietf.org/internet-drafts/draftietf-manet-aodv-13.txt

[5] D.Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” Mobile Computing, 1996.

[6] C. E. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in the Second IEEE Workshop o Mobile Computing systems and Applications, 1999.

[7] V.D.Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” *in Proceedings of the IEEE Conference on Computer Communications (INFOCOM), 1997.*

[8] Y.B.Ko and N. H. Vaidya, “Location-aided routing (lar) in mobile ad hoc networks*,” in Proceedings of the IEEE/ACM International Conference on Mobile Computing and Networking (MOBICOM’ 98), 1998.*

[9] S.Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, “A distance routing effect algorithm for mobility (dream),” *in Proceedings of the IEEE/ACM international Conference on Mobile Computing and Networking (MOBICOM’98), 1998, pp. 76–84.*

[10] M.Gunes¸ U. Sorges, and I. Bouazisi, "Ara - the ant-colony based routing algorithm for manets*," in Proceedings of the ICPP Workshop on Ad Hoc Networks. IEEE Computer Society Press, 2002,pp. 79–85.*

[11] J.S.Baras and H.Mehta, "A probabilistic emergent routing algorithm for mobile ad hoc networks," in Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2003.

[12] D. Camara and A. Loureiro, "A novel routing algorithm for ad hoc networks, "in *Proceedings of the 33rd Hawaii International Conference on System Sciences, 2002.*

[13] M. Heissenb¨uttel and T. Braun, "Ants-based routing in large scale mobile ad-hoc networks" Institute of Computer Science and Applied Mathematics, University of Berne, Tech. Rep. CH-3012, 2003.

[14] Z. J. Hass, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *in Proceedings of the IEEE Conference on Computer Communication, 2002.*

[15] G. Di Caro and M. Dorigo, "Antnet: A mobile agents approach to adaptive routing," IRIDIA, Universit´e Libre de Bruxelles, Tech. Rep. IRIDIA 97-12, 1997.

[16] The NS -2 Manual, the VINT Project.

[17] Jyoti Jain, M. Tech. Dissertation on "Optimization of AODV Heuristic Approach" under the guidance of Dr. Aditya Goel, MANIT (Deemed University), Bhopal, 2007.

[18] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Yetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," *in Proceedings of the ACM/IEEE International Conferenceon Mobile Computing and Networking (MOBICOM'98),1998*

[19] A Mobility Based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks A. Bruce McDonald and Taieb Znati, IEEE Journal on Selected Areas in Communications, Vol. 17, No. 8, August 1999

[20] Biologically Inspired Discrete Event Network Modelling by Hessam S. Sarjoughian. "The biological inspiration is provided by Honey-Bee Scout-Recruit System"

[21] Probability Routing Algorithm for Mobile Ad Hoc Networks' Resources Management Osama H. Hussein, Member, IEEE, Tarek N. Saadawi, and Myung Jong Lee, Senior Member, IEEE, vol. 23, no. 12, december 2005

[22] Aditya Goel, Ravi Shankar Mishra, "Remote Data Acquisition Using Wireless – Scada System" International Journal of Engineering (IJE), Volume (3): Issue (1)

[23] Aditya Goel, Er.R.K.Sethi, "Integrated Optical wireless Network for Next Generation Wireless Systems" Signal Processing: An International Journal (SPIJ), Volume (3): Issue (1)

Testing Various Similarity Metrics and their Permutations with Clustering Approach in Context Free Data Cleaning

Sohil D. Pandya sohilpandya@gmail.com
Lecturer, MCA Department
Sardar Vallabhbhai Patel Institute of Technology
Vasad, 388306, India

Dr. Paresh V. Virparia pvvirparia@yahoo.com
Reader, G H Patel PG Dept. of Comp. Sci. & Technology,
Sardar Patel University,
Vallabh Vidyanagr, 388120, India

Abstract

Organizations can sustain growth in this knowledge era by proficient data analysis, which heavily relies on quality of data. This paper emphasizes on usage of sequence similarity metric with clustering approach in context free data cleaning to improve the quality of data by reducing noise. Authors propose an algorithm to test suitability of value to correct other values of attribute based on distance between them. The sequence similarity metrics like Needlemen-Wunch, Jaro-Winkler, Chapman Ordered Name Similarity and Smith-Waterman are used to find distance of two values. Experimental results show that how the approach can effectively clean the data without reference data.

Keywords: Context free data cleaning, Clustering, Sequence similarity metrics.

1. INTRODUCTION

The management of any organizations is intense to sustain growth in markets and to achieve this; it heavily relies on data analysis. Data analysis provides an impending to proficient way for by and large picture and brings to float up detailed and hidden Information. The accuracy and relevance of data analysis heavily relies on the quality of data. The data quality measures like completeness, valid, consistent, timeliness, accurate, relevance etc. allow quantifying data in order to achieve high performance results of various data analyses. Because of human interventions and computations at various levels, noise is added in data before it got stored [3]. Noise is “irrelevant or meaningless data” [1], which leads to deterioration of outcome of data analysis. The data cleaning is a process of maintaining data quality in Information Systems (IS) [2]. The data cleaning processes mainly focused on detection and removal of noise. Using similarity metrics in data cleaning process to identify and replace incorrect sequence with correct sequence based on distance between them is an interesting way of cleaning the data [6]. Here the distance for various similarity metrics may be based numbers of characters, number of replacements needed to convert one sequence to another, number of re-arrangements required, most similar characters, any combinations of all above, etc. The distance between two sequence ranges from 0.0 to 1.0. For example, the distance between *Malaysia* and *Mallayssia* for various similarity metrics is shown Table 1.

The purpose of this paper is to apply similarity metrics and their permutations in *context free data cleaning* using *clustering* approach. The context free data cleaning means to examine and transform values of attributes without taking account further values of attribute [4].

Similarity Metrics	Distance
Needlemen-Wunch	0.8000
Smith-Waterman	0.8750
Chapman Ordered Name Compound Similarity	0.9375
Jaro-Winkler	0.9533

TABLE 1: Values of Distance for Various Similarity Metrics.

The core idea is based on the frequency of values and based on matching between them it is decided whether they should be transformed or not? Clustering is the assignment of a set of observations into subset so that observations in the same clusters are similar in some sense, which has various applications in machine learning, data mining, pattern recognition, bioinformatics, etc. [7]. The later sections describe an algorithm, its experimental results and concluding remarks.

2. USAGE OF SIMILARITY METRICS IN CONTEXT FREE DATA CLEANING

The proposed algorithm has two major components, viz. clustering and similarity, and one important parameter *acceptableDist*, which is minimum acceptable distance required during matching and transformation. To measure distance we used following Similarity Metrics and their permutations:

1. Needlemen-Wuch
2. Jaro-Winkler
3. Smith-Waterman
4. Champan Ordered Name Compound Similarity

The Needleman-Wunch algorithm, as in (1) performs a global alignment on two sequences and commonly used in Bioinformatics to align protein sequences [8].

$$F_{0j} = d * j$$

$$F_{i0} = d * i \qquad (1)$$

$$F_{ij} = \max(F_{i-1,j-1} + S(S_{1i}, S_{2j}), F_{i,j-1} + d, F_{i-1,j} + d$$

Where $S(S_{1i},S_{2j})$ is the similarity of characters i and j; d is gap penalty.

The Jaro-Winkler distance, as in (2), is the major of similarity between two strings [8]. It is a variant of Jaro distance [8].

$$Jaro-Winkler(S_1, S_2) = Jaro(S_1, S_2) + (L * p(1 - Jaro(S_1, S_2))$$

$$Jaro(S_1, S_2) = \frac{1}{3}\left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m}\right) \qquad (2)$$

Where m is number of matching characters and t is number of transpositions required; L is length of common prefix and p is scaling factor (standard value 0.1).

The Smith-Waterman algorithm, as in (3) is well-known algorithm for performing local sequence alignment, i.e. for determining similar regions between two protein sequences. It compares segments of all possible lengths and optimizes the similarity measures using substitution matrix and gap scoring scheme [8].

$$H(i,0) = 0, 0 \le i \le m$$

$$H(0,j) = 0, 0 \le j \le n \qquad (3)$$

$$H(i,j) = \max\left\{\begin{matrix} 0 \\ H(i-1,j-1) + w(S_{1i}, S_{2j}), Mismatch \\ H(i-1,j) + w(S_{1i}, -), Deletion \\ H(i,j-1) + w(-, S_{2j}), Insertion \end{matrix}\right\}$$

Where S1, S2 are strings and m, n are their lengths; H (i, j) is the maximum similarity between strings of S1 of length i and S2 of length j; w(c,d) represents gap scoring scheme.

Chapman Ordered Name Compound Similarity tests similarity upon the most similar terms of token-based name where later name are valued higher than earlier names [8].

2.1 Algorithm

***Step-1*:** Start.
***Step-2*:** Values for a selected attributed transformed into uppercase, after removal of non-alphanumeric characters.
***Step-3*:** Derive frequencies in descending order, for all the distinct sequences. Refer the group of distinct values as clusters and the sequences as cluster identifiers.
Step-4: Select any of the sequence similarity metrics for comparing two values of an attribute and decide acceptableDist.
***Step-5*:** Compare the cluster identifier with other cluster identifiers, beginning with first to last cluster, to decide distance between them.
***Step-6*:** If the distance is less than *acceptableDist* then it forms transformation and/or validation rules for particular *acceptableDist* that can be utilized in further cleaning process (e.g., second pass of the same algorithm, context dependant cleaning) and the values of comparables can be transformed in to comparator, else comparables remains as separate clusters.
***Step-7*:** Stop.

[Note: The extended version of the above algorithm is used for usage of various permutations of two Similarity Metrics, where we had two parameters – one for each Similarity Metrics, i.e. *accetableDist1* and *acceptableDist2* [5]. In the extended version we perform *Step-6* for both Similarity Metrics. The results for both approach is shown in Section 3]

2.2 Assumptions & Limitations

In the above experiments we made certain assumptions like (a) Most of data entry is done correctly, only 2 to 20 percent data injected is not correct, and (b) Entered incorrect values are typographic errors. The algorithm has limitations like (a) It may incorrectly altered values those may be correct in real world, (b) Correct values which are typographically similar may be transformed, (c) The result varies when same *acceptableDist* and different Similarity Metrics (or its combinations) upon a same dataset, which leads to confusion upon selection of Similarity Metrics.

3. EXPERIMENTAL RESULTS

The algorithm is tested using a sample data derived from Internet. The data consisting of attributes named First Name, Middle Name, Last Name, Address, City, Pin code, District, State, Country, Phone number, and Email. District attribute is selected the testing purpose. There were about 13,074 records out of which 551 (4.22 %) values for the selected attribute were identified as incorrect and required corrections. During the execution of algorithm, 359 clusters were identified for the selected attribute. After identification of clusters and their identifiers, algorithm is tested for various similarity metrics value. For selected similarity metrics various results, like how many records updated (total, correctly & incorrectly), were found and are discussed in this section. Following results, percentage of correctly altered (CA %), percentage of incorrectly altered (IA %) and percentages of unaltered values (UA %) were derived as in (4).

$$\begin{aligned} CA(\%) &= \frac{CA}{TotalAlteration} * 100 \\ IA(\%) &= \frac{IA}{TotalAlteration} * 100 \\ UA(\%) &= \frac{UA}{NumberofIncorrectValues} * 100 \end{aligned} \tag{4}$$

Results found on testing of algorithm are:

1. It can be observed from Figure 1 that application of Similarity Metrics and their permutations that the percentage of values altered is growing with increase of *acceptableDist* as the tolerance of matching criteria. (See Table 2 for Legend description). For instance, using Chapman Ordered Name Compound Similarity with distance values 1, 0.9, and 0.8 (of each) there were 0.35%, 4.60%, 9.81% values altered respectively out of all values.

2. It can also be observed from Figure 2 and 3 that as the increment of *acceptableDist*, the percentage of incorrectly altered values also gets increased. For instance, using Chapman Ordered Name Compound Similarity with distance values 1, 0.9, and 0.8 (of each) there were 7.14%, 38.57%, and 57.16% values altered incorrectly out of total values altered.

3. The efficiency of algorithm is increased, if we use permutation of Similarity Metrics instead of using a single Similarity Metric.

Sr. No.	Notation	Similarity Metric – I	Similarity Metric-II
1	NW	Needlemen-Wunch	-
2	JW	Jaro-Winkler	-
3	CONS	Chapman Ordered Name Compound Similarity	-
4	SW	Smith-Waterman	-
5	NWJW	Needlemen-Wunch	Jaro-Winkler
6	NWCONS	Needlemen-Wunch	Chapman Ordered Name Compound Similarity
7	NWSW	Needlemen-Wunch	Smith-Waterman
8	JWCONS	Jaro-Winkler	Chapman Ordered Name Compound Similarity
9	JWSW	Jaro-Winkler	Smith-Waterman
10	CONSSW	Chapman Ordered Name Similarity	Smith-Waterman

TABLE 2: Legend Description.

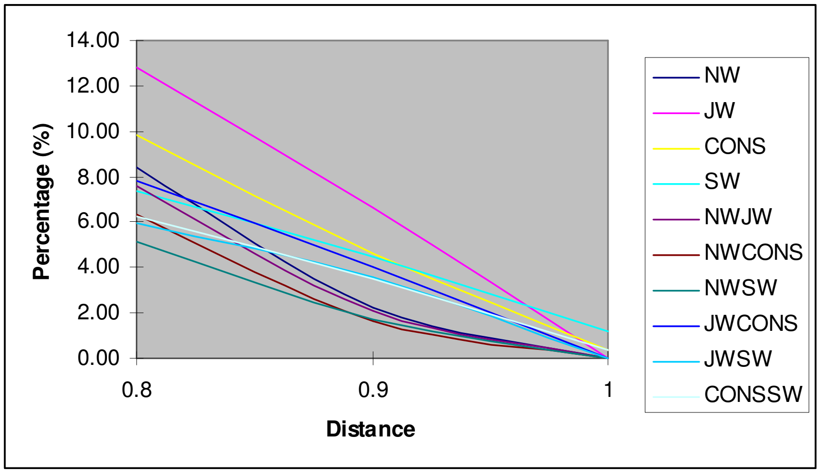


FIGURE 1: Percentage Alteration.

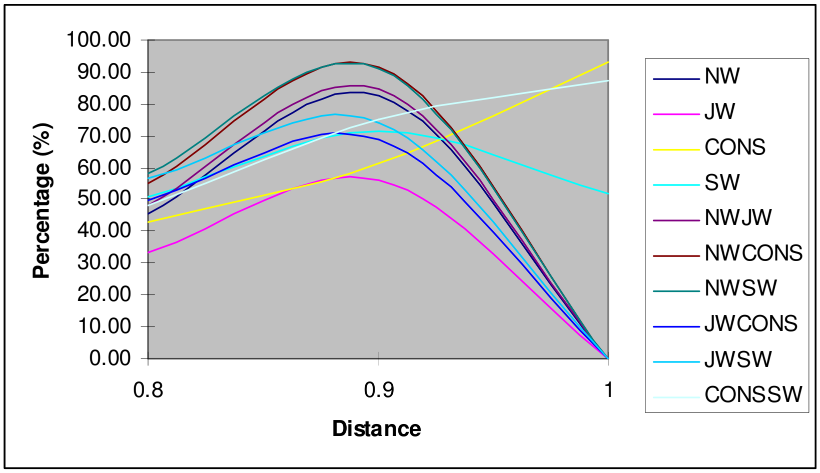


FIGURE 2: Percentage of Correctly Altered Values.

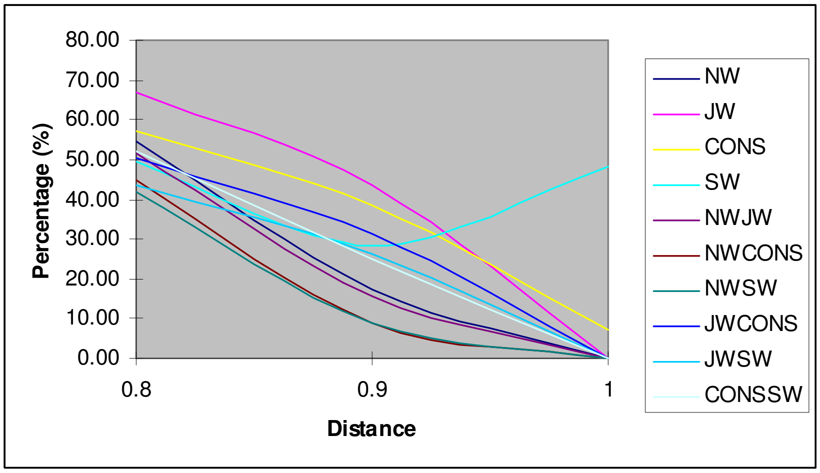


FIGURE 3: Percentage of Incorrectly Altered Values.

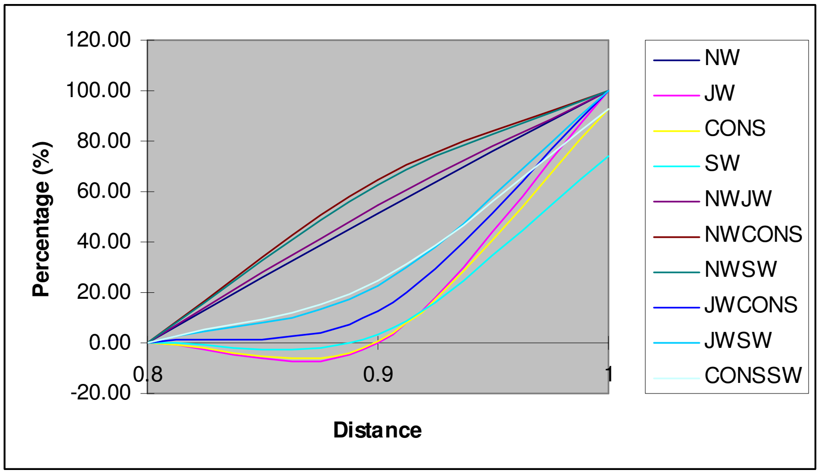


FIGURE 4: Percentage of Unaltered Values.

4. CONSLUSION

The results of the experiment verify the correctness of the algorithm and which motivate us to use it for data cleaning. The major advantage of it, where the reference/correct dataset is not given and still the data cleaning is achieved. However the various percentages shown in results depend on Similarity Metric(s), parameter(s), and dataset, i.e. for different dataset may require different aspects of said dependants. It may be possible that other Similarity Metrics or their permutations may give more precise data cleaning, that yet to be explored and future experiments.

5. REFERENCES

1. Hui Xiong, Gaurav Pandey, Michael Steinbach, Vipin Kumar. *"Enhancing Data Analysis with Noise Removal"*. IEEE Transaction on Knowledge & Data Engineering, 18(3):304-319, 2006.
2. Lukasz Ciszak. *"Applications of Clustering and Association Methods in Data Cleaning"*. In Proceedings of the International Multiconference on Computer Science and Information Technology. 2008.
3. Sohil D Pandya, Dr. Paresh V Virparia. *"Data Cleaning in Knowledge Discovery in Databases: Various Approaches"*. In Proceedings of the National Seminar on Current Trends in ICT, INDIA, 2009.
4. Sohil D Pandya, Dr. Paresh V Virparia. *"Clustering Approach in Context Free Data Cleaning"*. National Journal on System & Information Technology, 2(1):83-90, 2009.
5. Sohil D Pandya, Dr. Paresh V Virparia. *"Application of Various Permutations of Similarity Metrics with Clustering Approach in Context Free Data Cleaning"*. In Proceedings of the National Symposium on Indian IT @ CROXRoads, INDIA, 2009.
6. W Cohen, P Ravikumar, S Fienberg. *"A Comparison of String Distance Metrics for Name-Matching Tasks"*. In the Proceedings of the IJCAI, 2003.
7. http://en.wikipedia.org/
8. http://www.dcs.shef.ac.uk/~sam/simmetric.html

Mining Spatial Gene Expression Data Using Association Rules

M.Anandhavalli anandhigautham@gmail.com
Reader, Department of Computer Science & Engineering
Sikkim Manipal Institute of Technology
Majitar-737136, India

M.K.Ghose mkghose2000@yahoo.com
Prof&Head, Department of Computer Science & Engineering
Sikkim Manipal Institute of Technology
Majitar-737136, India

K.Gauthaman gauthamank@gmail.com
Prof&Head, Department of Pharmacognosy
Himalayan Pharmacy Institute
Majitar, East Sikkim-737136, India

Abstract

One of the important problems in data mining is discovering association rules from spatial gene expression data where each transaction consists of a set of genes and probe patterns. The most time consuming operation in this association rule discovery process is the computation of the frequency of the occurrences of interesting subset of genes (called candidates) in the database of spatial gene expression data. In this paper, an efficient method for mining strong association rules from spatial gene expression data is proposed and studied. The proposed algorithm adopts Boolean vector with relational AND operation for discovering the frequent itemsets without generating candidate itemsets and generating strong association rules with fixed antecedents. Experimental results show that the proposed algorithm is fast and memory efficient for discovering of frequent itemsets and capable of discovering meaningful association rules in effective manner.

Keywords: Spatial Gene expression data, Association Rule, Frequent itemsets, Boolean vector, Similarity Matrix.

1. INTRODUCTION

The main contribution here has been a great explosion of genomic data in recent years. This is due to the advances in various high-throughput biotechnologies such as spatial gene expression database. These large genomic data sets are information-rich and often contain much more information than the researchers who generated the data might have anticipated. Such an enormous data volume enables new types of analyses, but also makes it difficult to answer research questions using traditional methods. Analysis of these massive genomic data has two important goals:

1) To determine how the expression of any particular gene might affect the expression of other genes
2) To determine what genes are expressed as a result of certain cellular conditions, e.g. what genes are expressed in diseased cells that are not expressed in healthy cells?

The most popular pattern discovery method in data mining is association rule mining. Association rule mining was introduced by [4]. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in transaction databases or other data repositories. The relationships are not based on inherent properties of the data themselves but rather based on the co-occurrence of the items within the database. The associations between items are commonly expressed in the form of association rules. In this setting, attributes which represents items are assumed to have only two attributes and thus referred as Boolean attributes. If an item is contained in a transaction, the corresponding attribute value will be 1; otherwise the value will be 0. Many interesting and efficient algorithms have been proposed for mining association rules for these Boolean attributes, for examples, Apriori [3], DHP [6], and partition algorithms [7]. Currently most association mining algorithms are dedicated to frequent itemsets mining. These algorithms are defined in such a way that they only find rules with high support and high confidence. A characteristic of frequent itemsets mining is that it relies on there being a meaningful minimum support level that is sufficiently high to reduce the number of frequent itemsets to a manageable level. A huge calculation and complicated transaction process are required during the frequent itemsets generation procedure. Therefore, the mining efficiency of the Apriori-like algorithms is very unsatisfactory when transaction database is very large particularly spatial gene expression database.

In this paper, an attempt has been made to propose a novel, fast and memory efficient algorithm for discovering of frequent itemsets and for generating meaningful association rules in effective manner from spatial gene expression data.

2. MATERIALS AND METHODS

2.1 SPATIAL GENE EXPRESSION DATA

The Edinburgh Mouse Atlas gene expression database (EMAGE) is being developed as part of the Mouse Gene Expression Information Resource (MGEIR) [1] in collaboration with the Jackson Laboratory, USA. EMAGE (http://genex.hgu. mrc.ac.uk/Emage/database) is a freely available, curated database of gene expression patterns generated by in situ techniques in the developing mouse embryo. The spatial gene expression data are presented as N×N similarity matrix. Each element in the matrix is a measure of similarity between the corresponding probe pattern and gene-expression region. The similarity is calculated as a fraction of overlap between the two and the total of both areas of the images. This measurement is intuitive, and commonly referred to as the Jaccard index [2]. When a pattern is compared to itself, the Jaccard value is 1 because the two input spatial regions are identical. When it is compared to another pattern, the Jaccard Index will be less than one. If the Jaccard Index is 0, the two patterns do not intersect. If a Jaccard Index value is close to 1, then the two patterns are more similar.

However, biologists are more interested in how gene expression changes under different probe patterns. Thus, these similarity values are discretized such that similarity measure greater than some predetermined thresholds and converted into Boolean matrix.

2.2 DATA PREPROCESSING

Preprocessing is often required before applying any data mining algorithms to improve performance of the results. The preprocessing procedures are used to scale the data value either 0 or 1. The values contained in the spatial gene expression matrix had to be transformed into Boolean values by a so-called discretization phase. In our context, each quantitative value has given rise to the effect of four different discretization procedures [2]: Max minus x% method, Mid-range-based cutoff method, x% cut off and x% of highest value method.

Max minus x% procedure consists of identifying the highest expression value (HV) in the data matrix, and defining a value of 1 for the expression of the gene in the given data when the expression value was above HV – x% of HV where x is an integer value. Otherwise, the expression of the gene was assigned a value of 0 (Figure 1a).

Mid-range-based cutoff (Figure 1b) identifies the highest and lowest expression values in the data matrix and the mid-range value is defined as being equidistant from these two numbers (their arithmetic mean). Then, all expression values below or equal to the mid-range were set to 0, and all values strictly above the mid-range were set to 1.

x% of highest value approach (Figure 1c) identifies data in which its level of expression is in the 5% of highest values. These are assigned the value 1, and the rest were set to 0.

Value greater than x% approach (Figure 1d) identifies the level of expression and assigns the value 1 when it is greater than given percentage and the rest are set to 0.

From these four different procedures resulted in different matrix densities, the first and last procedure resulted in the same number of Boolean 1 results for all gene expressions, whereas the second and fourth procedure generated same densities of 1, depending on the gene expression pattern throughout the various data matrix.

From the similarity matrix, two different sets of transactions are constructed, which in turn lead to two different types of association rules.

1. The items I are genes from the data set, where a transaction T ⊆ I consists of genes that all have an expression pattern intersecting with the same probe pattern.

2. The items I are the probe patterns, where a transaction T ⊆ I consists of probe patterns all intersecting with the expression patterns in the same image.

To create the first type of transactions, we take for each probe pattern r, every gene g from which its associated gene expression pattern g satisfies the minimum similarity β, i.e., similarity(r, g) > β, to form the itemsets.
The second type of transactions is created in a similar way. For each gene expression pattern g in the database we create an itemsets that consists of a set of probe patterns that intersect with the gene expression pattern g. Each probe pattern r must satisfy the minimum similarity β, i.e.., similarity(r, g) > β, to get included in the itemsets.

	α (Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	0
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE 1a: Results of Max minus 25% method

	α (Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	0
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE 1b: Results of Mid-range-based cutoff

	α(Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	1
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE 1c: Results of x% of highest value approach

	α(Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	1
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE1d: Results of Value greater than x% approach

FIGURE 1: Schematic description of the discretization protocols used

2.3 ASSOCIATION RULE MINING

The Apriori-like algorithms adopt an iterative method to discover frequent itemsets. The process of discovering frequent itemsets need multiple passes over the data. .The algorithm starts from frequent 1-itemsets until all maximum frequent itemsets are discovered. The Apriori-like algorithms consist of two major procedures: the join procedure and the prune procedure. The join procedure combines two frequent k-itemsets, which have the same (k-1)-prefix, to generate a (k+1)-itemset as a new preliminary candidate. Following the join procedure, the prune procedure is used to remove from the preliminary candidate set all itemsets whose k-subset is not a frequent itemsets [3].
From every frequent itemset of k>=2, two subsets A and C, are constructed in such a way that one subset C, contains exactly one item in it and remaining k-1 items will go to the other subset A. By the downward closure properties of the frequent itemsets these two subsets are also frequent and their support is already calculated. Now these two subsets may generate a rule A →C, if the confidence of the rule is greater than or equal to the specified minimum confidence.

2.4 ALGORITHM DETAILS

[1] Let $I=\{i_1, i_2, \dots, i_n\}$ be a set of items, where each item ij corresponds to a value of an attribute and is a member of some attribute domain $Dh=\{d_1, d_2, \dots, d_s\}$, i.e. $i_j \in D_h$. If I is a binary attribute, then the Dom (I)={0,1}. A transaction database is a database containing transactions in the form of (d, E), where d $\in$ Dom(D) and E $\in$ I.

[2] Let D be a transaction database, n be the number of transactions in D, and minsup be the minimum support of D. The new_support is defined as new_support = minsup × n.

[3] Proposition 1: By Boolean vector with AND operation, if the sum of ‘1’ in a row vector Bi is smaller than k, it is not necessary for Bi to involve in the calculation of the k- supports.

[4] Proposition 2: According to [5], Suppose Itemsets X is a k-itemsets; |FK-1(j)| presents the number of items ‘j’ in the frequent set F_{K-1}. There is an item j in X. If | F_{K-1}(j)| is smaller than k-1, itemset X is not a frequent itemsets.

[5] Proposition 3: |F_K| presents the number of k-itemsets in the frequent set F_K. If |F_K| is smaller than k+1, the maximum length frequent itemsets is k.

[6] Lemma 1: If there exists two rules A→B and A→ {B U X}, where $X \not\in AUB$, then the confidence of the second cannot be larger than first one.

The proposed algorithm for finding the association rules in terms of spatial gene expression data in the form of similarity matrix consists of five phases as follows:

1. Transforming the similarity matrix into the Boolean matrix
2. Generating the set of frequent 1-itemsets F_1
3. Pruning the Boolean matrix
4. Generating the set of frequent k-itemsets F_k(k>1)
5. Generating association rules from the generated frequent itemsets with confidence value greater than a predefined threshold (minconfidence).

A detailed description of the proposed algorithm is described as follows:

Part 1: Algorithm for generating frequent itemsets

Input: Spatial Gene Expression data in similarity matrix (M), the minimum support.
Output: Set of frequent itemsets F.

```
Normalize the data matrix M and transformed into Boolean
  Matrix B;
 // Frequent  1-itemset generation
For each column Ci of B
  If sum($C_i$) >= new_support
     F1 =  { $I_i$};
  Else delete $C_i$ from B;
 // By Proposition 1
For  each row $R_j$ of B
  If sum($R_j$) < 2
   Delete  $R_j$ from B;
 // By Proposition 2  and 3
For (k=2; | $F_{k-1}$| > k-1; k++)
{
  // Join procedure
   Produce k-vectors combination for all columns of B;
   For each k-vectors combination { $B_{i1}$, $B_{i2}$,…$B_{ik}$}
   {  E=  $B_{i1} \cap B_{i2} \cap \dots \cap B_{ik}$
      If sum(E) >= new_support
     Fk = { $I_{i1}$, $I_{i2}$,…$I_{ik}$}
   }
  // Prune procedure
 For each item $I_i$ in $F_k$
   If  |$F_k(I_i)$| < k
     Delete the column Bi according to item Ii from B;
 For each row $R_j$ of  B
   If sum($B_j$) < k+1
      Delete Bj from B;
    k=k+1
 }
Return F = $F_1 U F_2 \dots U F_k$
```

This algorithm is capable of discovering all possible set of frequent itemsets subject to a user specified minimum confidence.

Part 2: Algorithm for generating association rules.

Input: Set of Frequent (F) with descending order of new_support count and minimum confidence.
Output: Set of Association rules

```
For all $f_k$, $f_k \in$ F, k=1 to max_size-1  do
{
req_support= new_support($f_k$) X minconfidence
total=0
for all $F_m$ , $F_m \in$ F, m=k+1 to max_size do
   {
   if new_support($F_m$) >= req_support then
      {
// By lemma 1
10.        If ($F_k \subseteq F_m$) then
11.            {
12.              total =totoal+1
13.              conf= new_support( $F_m$)/new_support($F_k$)
14.              Generate the rule $F_k \longrightarrow (F_m-F_k)$ &=conf and new_support=new_support($F_m$)
15.            }
16.         else
17.           If ( total < 2) continue step1 with next k
18.          else
19.             total=0
20.         }
21.      }
22.  }
```

This algorithm is capable of finding all association rules with a fixed antecedent and with different consequents from the frequent itemsets subject to a user specified minimum confidence very quickly. The proposed algorithm is avoiding the unnecessary checking for the rules based on the above lemma 1.The algorithm generate the rules with a fixed antecedent part. When all the rules with that antecedent are generated it will go to the next antecedent. For a given antecedent if all rules in the level, where k is the number of items in the consequent, have confidence less than the threshold, i.e. no rules are generated, and then the confidence of any rule in k+1 level also cannot be more than threshold. So checking for rules from this level onward can be avoided without missing any rules. Now the maximum possible confidence of the rule in the k+1 level will be minimum confidence of the two itemsets from which this is constructed. Since the confidence of only one of them is larger than the threshold, others must be less than the threshold. So the confidence of the rule in k+1 will be less than threshold. So, it is not necessary to check for the rules in the next level without missing any valid rule. So it can be concluded that the proposed algorithm is complete.

3. RESULTS AND DISCUSSION

The proposed algorithm was implemented in Java and tested on Linux platform. Comprehensive experiments on spatial gene expression data has been conducted to study the impact of normalization and to compare the effect of proposed algorithm with Apriori algorithm. Figure 2 and 3 gives the experimental results for execution time (generating frequent itemsets and finding rules) vs. user specified minimum supports and shows that response time of the proposed algorithm is much better than that of the Apriori algorithm. In this case, confidence value is set 100% for the rule generation, which means that all the rules generated are true in 100% of the cases.

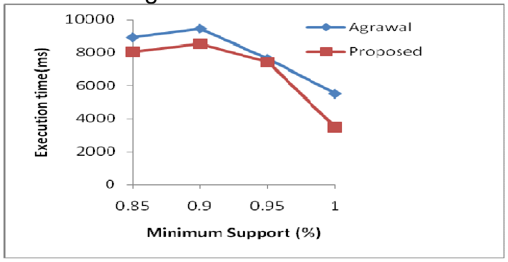


FIGURE 2: Performance on Stage 14 of EMAGE Spatial Gene expression data (Minsupport vs. Execution time)

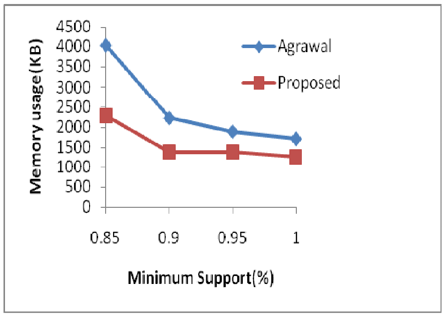


FIGURE 3: Performance on Stage 17 of EMAGE Spatial Gene expression data (Minsupport vs. Execution time)

Figure 4 and 5 gives the experimental results for memory usage vs. user specified minimum supports and results show that proposed algorithm uses less memory than that of Apriori algorithm because of the Boolean and relational AND bit operations.

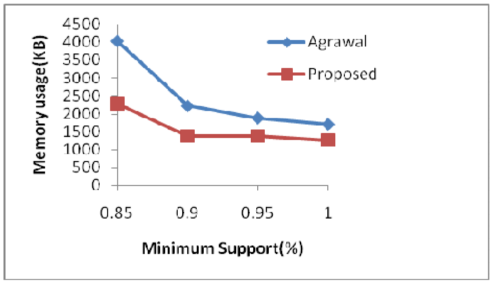


FIGURE 4: Performance on Stage 14 of EMAGE Spatial Gene expression data (Minsupport vs. Memory usage)

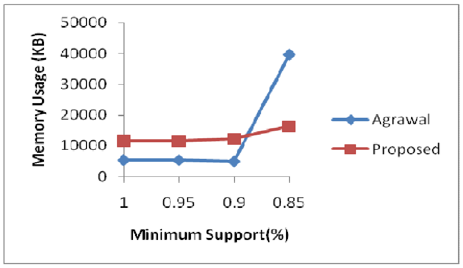


FIGURE 5: Performance on Stage 17 of EMAGE Spatial Gene expression data (Minsupport vs. Memory usage)

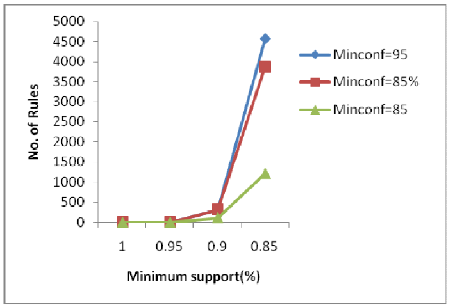


FIGURE 6: Association rules and Minsup in Apriori algorithm Stage 14 of EMAGE Spatial Gene expression

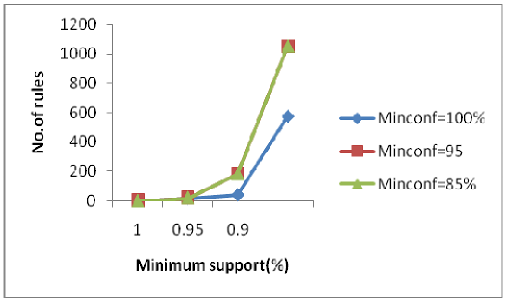


FIGURE 7: Association rules and Minsup in Proposed algorithm Stage 14 of EMAGE Spatial Gene expression

The number of association rules decreases along with an increase in minimum support under a given specific minimum confidence, which shows an appropriate Minsupport (or Minconfidence) can constraint the number of association rules and avoid the occurrence of some association rules so that it cannot yield a decision. These results have shown in Figures 6-7 for the Stage 14 of EMAGE spatial gene expression data. The results are as expected and quite consistent with our intuition.

4. CONCLUSION

In this paper, a novel method of mining frequent itemsets and strong association rules from the spatial gene expression data has been proposed to generate frequently occur genes very quickly. The proposed algorithm does not produce candidate itemsets, it spends less time for calculating k-supports of the itemsets with the Boolean matrix pruned, and it scans the database only once and needs less memory space when compared with Apriori algorithm. The proposed algorithm is good enough for generating association rules from spatial gene expression data and it is very fast and memory efficient. Finally, the large and rapidly increasing compendium of data demands data mining approaches, particularly association rule mining ensures that genomic data mining will continue to be a necessary and highly productive field for the foreseeable future.

5. ACKNOWLEDGMENT

This study has been carried out as part of Research Promotion Scheme (RPS) Project under AICTE, Govt. of India.

6. REFERENCES

1. Baldock,R.A., Bard,J.B., Burger,A., Burton,N., Christiansen,J., Feng,G., Hill,B., Houghton,D., Kaufman,M., Rao,J. et al., “EMAP and EMAGE: a framework for understanding spatially organized data”, Neuroinformatics, 1, 309–325, 2003.
2. Pang-Ning Tan, Micahel Steinbach, Vipin Kumare, ”Intoduction to Data Mining Pearson Education”, second edition, pp.74, 2008.
3. Agrawal, R. & Srikant, R., “Fast Algorithms for Mining Association Rules in large databases”. In Proceedings of the 20th International Conference on Very Large Databases pp. 487-499. Santiago, Chile, 1994.
4. Agrawal, R., Imielinski, T., & Swami, A., ”Mining association rules between sets of items in large databases”. Proceedings of the ACM SICMOD conference on management of data”, Washington, D.C, 1993.
5. Xu, Z. & Zhang, S., “An Optimization Algorithm Base on Apriori for Association Rules”. Computer Engineering 29(19), 83-84, 2003.
6. J S. Park and M -S. Chen and PS. Yu, “An effective hash-based algorithm for mining association rules”, Proceedings of the ACM SIGMOD International Conference on Management of Data", San Jose, CA, May 1995.
7. A Savasere, E. Ommcinskl and S Navathe, “An efficient algorithm for mining association rules in large databases”, In Proceedings Of the 21st International Conference on Very Large Databases, Zurich, Switzerland, September 1995.

PERFORMANCE IMPROVEMENTS AND EFFICIENT APPROACH FOR MINING PERIODIC SEQUENTIAL ACCESS PATTERNS

D. Vasumathi
*Associate Professor, Department of Computer Science and Engineering,
JNTU College of Engineering.JNTU,
Kukatpally, Hyderabad-500 085,* *Vasukumar_devara@yahoo.co.in*
Andhra Pradesh, India.

Dr. A. Govardhan
*Principal, JNTU College of Engineering
JNTU,Jagityala, Karimnagar(Dist),
Andhra Pradesh, India*

K.Venkateswara Rao
*Associate Professor, Department of Computer Science and Engineering,
JNTU College of Engineering.JNTU,
Kukatpally, Hyderabad-500 085,
Andhra Pradesh, India.*

Abstract

Surfing the Web has become an important daily activity for many users. Discovering and understanding web users' surfing behavior are essential for the development of successful web monitoring and recommendation systems. To capture users' web access behavior, one promising approach is web usage mining which discovers interesting and frequent user access patterns from web usage logs. Web usage mining discovers interesting and frequent user access patterns from web logs. Most of the previous works have focused on mining common sequential access patterns of web access events that occurred within the entire duration of all web access transactions. However, many useful sequential access patterns occur frequently only during a particular periodic time interval due to user browsing behaviors and habits. It is therefore important to mine periodic sequential access patterns with periodic time constraints. In this paper, we propose an efficient approach, known as TCSMA (Temporal Conditional Sequence Mining Algorithm), for mining periodic sequential access patterns based on calendar-based periodic time constraint. The calendar-based periodic time constraints are used for describing real-life periodic time concepts such as the morning of every weekend. The mined periodic sequential access patterns can be used for temporal-based personalized web recommendations. The performance of the proposed TCSMA is evaluated and compared with a modified version of Web Access Pattern Mine for mining periodic sequential access patterns.

Keywords: Periodic Sequential Access Patterns, Web Access Patterns, Association Rule, Web Log Mining, TCSM&WAPM Algorithm

1. INTRODUCTION

With the explosive growth of resources available on the Internet, web surfing for useful information has become an important daily activity for many users. Web users surf for different web resources according to their needs, knowledge and interests. The discovery and understanding of web users' surfing habits are essential for the development of successful web monitoring and recommendation systems.

Web usage mining aims to discover interesting and frequent user access patterns from web usage data. The discovered knowledge is very useful for modeling users' web access behavior. Statistical techniques have traditionally been used for extracting statistical information from web logs. Association rule mining and sequential pattern mining have also been used to mine association and sequential patterns from web logs for web user access behavior. We can also visualize association and sequential patterns using WUM. However, most of these techniques do not consider the temporal aspect of access patterns.

Web usage mining[1], also known as web log mining, aims to discover interesting and frequent user access patterns from web browsing data that are stored in web server logs, proxy logs or browser logs. The discovered knowledge can be used for many practical applications such as web recommendations, adaptive web sites, and personalized web search and surfing. Many approaches[2-5] have been proposed for discovering sequential patterns from transaction databases. However, most of the pervious works only focused on mining common sequential access patterns of web access events, which occurred frequently within the entire duration of all web access transactions. In practice, many useful sequential access patterns occur frequently only in particular periodic time interval such as the morning of every weekend, but not in other time intervals due to user browsing behaviors and habits. Such sequential access patterns are referred to a periodic sequential access patterns, where periodic time intervals are real-life time concepts such a year, monthly, week and day. With periodic sequential access patterns, we can recommend or predict the occurrence of a web page during a particular time interval.

Recently, temporal association rule mining algorithms[6-8] have been proposed for mining temporal web access patterns. These works have discussed different ways for defining time constraints. However, such algorithms are mainly based on association rules that ignore the sequential characteristics of web access patterns. In addition, these algorithms also encounter the same problem as most Apriori-based algorithms that require expensive scans of database in order to determine which of the candidates are actually frequent. Different from temporal association rule mining, we propose an efficient approach, known as TCSMA (Temporal Conditional Sequence Mining Algorithm), to mine periodic sequential access patterns from web access transaction databases. We also define calendar-based periodic time constraints, which can be used for describing real-life time concept[9].

The rest of this paper is organized as follows. In Section2, we discuss calendar based periodic time constraints. The proposed TECMA is presented in Section3. the experimental results are shown in Section4. finally, the conclusions are given in Section5.

2 CALENDAR-BASED PERIODIC TIME CONSTRAINTS

In this section, we define calendar-based periodic time constraints, which can be used for describing real-life time concepts. The calendar-based periodic time constraints consist of calendar template and calendar instance.

Definition 2.1

A calendar template is defined as $CB_T = (CU_1\ I_1, CU_2\ I_2, \ldots, CU_n\ I_n)$.
Each CU_i is a calendar unit such as year, month, week, day, etc. and I_i is a closed interval that contains all valid time values (positive integers) of CU_i. A calendar template represents a hierarchy of calendar units and valid time intervals. For example, a typical calendar template can be in the form of (year [2007, 2008], month[1, 12], day [1, 31]) or (day-of-week [1, 7], hour [0, 3]).

Definition 2.2

Given a calendar template $CB_T = (CU_1\ I_1, CU_2\ I_2, \ldots, CU\ _n\ I_n)$, a calendar instance is denoted as $(I_{1'}, I_2', \ldots, I_n')$, where I_l ' is an nonempty set of positive integers and $I_l\ ' \subset I\ _i$, or is a wild-card symbol * that represents all valid time values in I_i. Calendar instances are formed from calendar template by setting some calendar units to specific values. It can then be used for describing real-life time concepts. For **example**,

Given CB_T = (day-of-week [1, 7], hour [0, 23]), we can have
C_I = ({6, 7}, {5, 6, 7, 8}) for the early morning of every weekend or
C_I = (*, {19, 20, 21}) for the evening of everyday.

In practice, some real-life time concepts such as morning or evening may have different meanings to different people depending on their personal behaviors and habits. For example, some people consider that morning is from sunrise to noon, while others consider that it is from 5 AM to 9 AM. Therefore, calendar instances can be defined according to actual practical requirements. We list some special calendar instances based on CB_T = (day-of-week [1, 7], hour [0, 23]) in Table 1.

Time Concept	Calendar Instances
Early morning	(*,{5.6.7.8})
Morning	(*,{9,10,11})
Noon	(*,{12})
Afternoon	(*,{13,14………17})
Evening	(*,{18,19,20,21})
Night	(*,{22,23,0…….4})
Weekdays	({1,2,………5}), *)
Weekend	({6,7}), *)

TABLE 1: Some special calendar instances.

Definition 2.3

A calendar-based periodic time constraint denoted as (C) = [CB_T, C_I] Where CB_T = calendar based template and C_I = one calendar instance For **example**, C = [(day-of-week [1, 7], hour [0, 23]), ({6, 7}, {8, 9})] represents “8:00 AM to 9:59 AM of every weekend”.

Given C = [CB_T, C_I], we say time t is covered by C If t belongs to the time interval defined by C. For **example**, t_1 = "2007-11-08 08: 22:45 Saturday" and t_2 = "2007-11-02 09:45:30 Sunday" are covered by C. If we denote the calendar-based periodic time constraint Call = [CB_T, (*, ..., *)], Where CB_T is any calendar based template, then it will specify all the time intervals.

3 THE TCSMA (TEMPORALCONDITIONALSEQUENCE MINING ALGORITHM)

In this section, we discuss our proposed approach, known as TCSMA (Temporal Conditional Sequence mining algorithm), for mining common and periodic sequential access patterns from a given web access transaction database.

3.1 Problem Statement

Generally, web logs can be regarded as a collection of sequences of access events from one user or session in timestamp ascending order. Preprocessing tasks [9] including data cleaning, user identification, session identification and transaction identification can be applied to the original web log files to obtain the web access transactions.

Let UAE = A set of unique access events, (which represents web resources accessed by
users, i.e. web pages, URLs, or topics) WAS = A web access sequence
WAS = $e_1e_2…e_n$ ($e_i \in$ UAE for $1 \le i \le n$) is a sequence of access events, and
|WAS| = n is called the length of WAS. Note that it is not necessary that $e_i \neq e_j$ for I $\neq$ j in
WAS, that is repeat of items is allowed
WAT = A web access transaction
WAT = (t, WAS), consists of a transaction time t and a web access sequence WAS

All the web access transactions in a database can belong to either a single user (for client-side logs) or multiple users (for server-side and proxy logs). The proposed algorithm does not depend on the type of web logs that contains the web access transactions. Suppose we have a set of web access transactions with the access event set, UAE = {a, b, c, d, e, f}. A sample web access transaction database is given in Table 2.

Transaction Time	Web Access Sequence
2007-11-03 20:21 : 10 Saturday	abdac
2007-11-04 21:45 : 22 Sunday	eaebcac
2007-11-07 18:23 : 24 Wednesday	cacbb
2007-11-10 21:10 : 10 Saturday	babfae
2007-11-10 21:30 : 20 Saturday	afbacfc

TABLE2: A database of web access transactions

WAS = A web access sequence and WAS = $e_1e_2…e_k\ e_{k+1}…e_n$,
WAS $_{prefix}$ = $e_1e_2…e_k$ is called a prefix sequence of WAS, or a prefix sequence of e_{k+1} in WAS.And
WAS $_{suffix}$ = $e_{k+1}e_{k+2}…$en is called a suffix sequence of WAS or a suffix sequence of e_k in WAS.
Now A web access sequence (WAS) = WAS $_{prefix}$ + WAS $_{suffix.}$
For **example**,
WAS = abdac can be denoted as WAS = a+bdac = ab+dac = … = abda+c.

Let S_1 and S_2 be two suffix sequences of e_i in WAS, and S_1 is also the suffix sequence of e_i in S_2. Then S_1 is called the sub-suffix sequence of S_2 and S_2 is the super-suffix sequence of S_1. The suffix sequence of e_i in WAS without any super-suffix sequence is called the long suffix sequence of e_i in WAS. For **example**, if WAS = abdacb, then S_1 = cb is the sub-suffix sequence of S_2 = bdacb and S_2 is the super-suffix sequence of S_1. S_2 is also the long suffix sequence of a in WAS.

Given

WAT_{DB}= A web access transaction database

$WAT_{DB} = \{(t1, S_1), (t2, S_2), \ldots, (t_m, S_m)\}$ in which WAS_i $(1 \le i \le m)$ is a web access sequence, and t_i is a transaction time.

Given a calendar-based periodic time constraint C that is defined in Section 2.

WAT_{DB} (C) = $\{(t_i, WAS_i) \mid t_i$ is covered by C, $1 \le I \le m\}$ is a subset of WAT_{DB} under C.

| WAT_{DB} (C)| is called the length of WAT_{DB} under C. The support of WAS in WAT_{DB} under C is defined in equation (3.1).

$$\mathrm{Sup(WAS,C)}= \frac{\left| \{S_i \mid WAS \in S_i,(t_i,S_i) \in WAT_{DB}(C)\} \right|}{\left| WAT_{DB}(C) \right|} \qquad 3.1$$

A web access sequence WAS is called a periodic sequential access pattern,
if sup(WAS, C) ≥MinSup, where MinSup is a given support threshold.

Let's consider the sample database in Table 2.

Suppose MinSup = 75% and calendar-based periodic time constraint

C =[(day-of-week [1, 7], hour [0, 23]), ({6, 7}, {20, 21})].

It is required to find all web access patterns supported by at least 75% access sequences within the time interval from 8:00 PM to 9:59 PM of every weekend from the sample database. If we use Call as the calendar-based periodic time constraint, the mining results should be all common sequential access patterns satisfying the given support threshold.

3.2 Proposed Approach

As shown in Figure:1, the proposed TCSMA consists of the following steps:

(1) Constraint Preprocessing;
(2) Constructing Event Queues for Conditional Sequence Base;
(3) Single Sequence Testing for Conditional Sequence Base;
(4) Constructing Sub-Conditional Sequence Base; and
(5) Recursive Mining for Sub-Conditional Sequence Base.

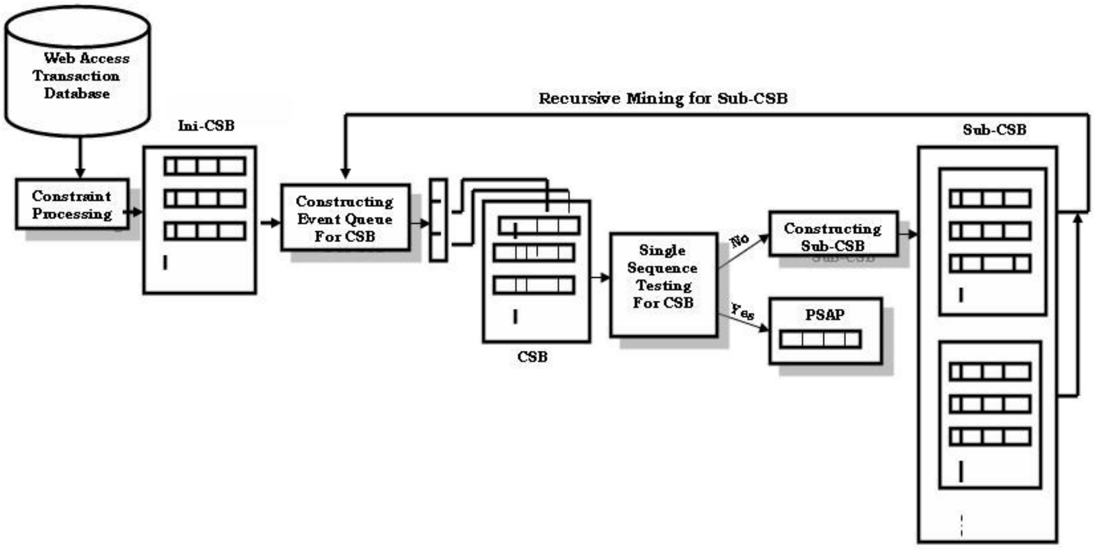

Web Access Transaction Database
Ini-CSB
Constraint Processing
Constructing Event Queue For CSB
Recursive Mining for Sub-CSB
CSB
Single Sequence Testing For CSB
No
Yes
Constructing Sub-CSB
PSAP
Sub-CSB

FIGURE 1: Overview of the proposed TCSMA

3.2.1 Constraint Preprocessing

The first step in the TCSMA is to filter the web access transaction database by discarding all transactions that do not satisfy the given calendar-based periodic time constraint. The remaining constraint-satisfied (S_c) transactions are then used to construct the initial conditional sequence base. The initial conditional sequence base and conditional sequence base are defined as follows.

Definition 3.1

The initial conditional sequence base, denoted as Ini-CSB, is the set of all constraint-satisfied transactions in the given web access transaction database, where constraint-satisfied transactions are web access transactions whose transaction times are covered by the given calendar-based periodic time constraint.

Definition 3.2

The conditional sequence base of an event e_i based on prefix sequence WAS_{prefix}, denoted as CSB(Sc), where

Sc = $WAS_{prefix} + e_i$, is the set of all long suffix sequences of e_i in sequences of a certain dataset.

If WAS_{prefix} = Ø, the dataset is equal to the initial conditional sequence base of the given web access transaction database. Otherwise, it is the conditional sequence base CSB(WAS_{prefix}).

We also call CSB (WAS c) the conditional sequence base of conditional prefix Sc. The initial conditional sequence base can also be denoted as CSB(Ø), with Sc =Ø. The ConsPreprocessing algorithm for constraint preprocessing of transactions from the web access transaction database WAT_{DB} is given in Figure: 2.

```
Algorithm: Cons Preprocessing

Input:
1: $C = [CB_T, C_I]$ – calendar-based periodic time constraint that consists of calendar based template $CB_T$ and calendar instance $C_I$
2: $WAT_{DB} = \{WAT_i \mid WAT_i = (t_i, WAS_i), 1 \le i \le n\}$ – web access transaction database, and $WAT_i$ is a web access transaction that consists of transaction time $t_i$ and web access sequence $WAS_i$
Output:
1: Ini-CSB - initial conditional sequence base of $WAT_{DB}$
Method:
1: Initialize Ini-CSB = $\emptyset$.
2: For each $WAT_i \in WAT_{DB}$, if $t_i$ is covered by C, insert $WAS_i$ into Ini-CSB.
3: Return Ini-CSB.
```

FIGURE 2: The algorithm for constraint preprocessing of transactions.

Example: Given a calendar-based periodic time constraint
C = [(day-of-week [1, 7], hour [0, 23]), ({6, 7}, {20, 21})], as the time of the third transaction in Table 3.2 is “2007-11-05 18:23:24 Wednesday”, it is not covered by C. So the web access sequence bbcac is discarded. After preprocessing, the Ini-CSB of the sample database contains {abdac, eaebcac, babfae, afbacfc}.

3.2.2 Constructing Event Queues for Conditional Sequence Base

The second step of the TCSMA is to construct event queues for CSB(Sc) (for Ini-CSB, Sc = Ø). The process performs the following four steps:

(1) Finding conditional frequent events from CSB(Sc);
(2) Creating a Header Table;
(3) Constructing event queues; and
(4) Deleting non-frequent events.

The conditional frequent event is defined as follows.

Definition 3.3

The conditional frequent event is the event whose support in the given conditional sequence base is not less than the support threshold, MinSup. To find conditional frequent events in CSB(Sc), we need to identify those events with support of greater than or equal to MinSup. This is given in equation (3.2) below.

$$\mathrm{Sup}(e_i) = \frac{\left|\{S_j \mid e_j \in S_j, S_j \in CSB(S_c)\}\right|}{\left|\text{Ini-CSB}\right|} \ge \mathrm{MinSup} \qquad 3.2$$

In equation (3.2), |{Sj | ei ∈ Sj, Sj ∈ CSB(Sc)}| is the number of sequences which contains the item labeled ei in CSB(Sc), and |Ini-CSB| is the length of Ini-CSB. Then, all the conditional frequent events form the entire Header Table of CSB(Sc). A linked-list structure for each conditional frequent event e_i, called e_i–queue, is created. Each item of e_i–queue is the first item labeled e_i in sequences of CSB(Sc). The head pointer of each event queue is recorded in the

Header Table. Finally, as all the items of sequences in CSB(Sc) which are labeled as non-frequent events are not needed anymore, they are discarded. The ConstructEQ algorithm for constructing event queues for CSB(Sc) is given in Figure:3

```
Algorithm: Construct EQ

Input:
1: MinSup - support threshold
2: CSB(Sc) - conditional sequence base of Sc
3: UAE = {ei|1 ≤ i ≤ n} – all access events in CSB(Sc)
Output:
1: CSB(Sc) with Header Table HT and event queues
Method:
1: Create an empty Header Table HT for CSB(Sc).
2: For each $e_i \in$ UAE, if sup($e_i$) ≥MinSup, insertn $e_i$ into HT.
3: For each conditional sequence $\in$ CSB(Sc) do
a) For each $e_i \in$ HT, insert the first item labeled ei in this sequence into $e_i$ -queue.
b) Delete all items of events $\notin$ HT from this sequence.
4: Return CSB(Sc) with HT and event queues.
```

FIGURE 3: The algorithm for constructing event queues for CSB.

Example For the Ini-CSB = {abdac, eaebcac, babfae, afbacfc}, the results after constructing the Header Table and event queues is given in Figure:4 Each access event is denoted as (event: count), where event is the event name and count is the number of sequences which contains the item labeled as event in Ini-CSB. To be qualified as a conditional frequent event (with MinSup = 75% and |Ini-CSB| = 4), an event must have a count of at least 3. Therefore, the conditional frequent events are (a:4), (b:4) and (c:3). The a-queue, b-queue and c-queue are shown by the dashed lines starting from the Header Table. The items labeled as non-frequent events d, e and f in each sequence are deleted. Similarly, for any subsequent conditional sequence base, the Header Table and event queues can also be constructed using the ConstructEQ algorithm.

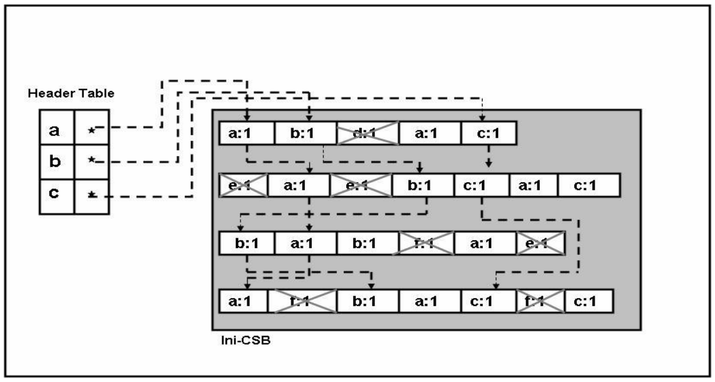


FIGURE 4: Ini-CSB with the Header Table and event queues.

3.2.3 Constructing Sub-Conditional Sequence Base

The sub-conditional sequence base is defined as follows.

Definition 3.4

CSB(WAS $_{prefix}$ +e $_i$) is called the sub-conditional sequence base of CSB(WAS $_{prefix}$), if $e_i \neq Ø$ for each access event e_i in the Header Table of CSB(Sc), the ConstructSubCSB algorithm for constructing CSB(Sc+e_i) based on CSB(Sc) is given in Figure:5

Algorithm: ConstructSubCSB

```
Input:
1: CSB(Sc) - conditional sequence base of Sc
2: $e_i$ - a given event in Header Table of CSB(Sc)
Output:
1: CSB(Sc+$e_i$ ) - conditional sequence base of $e_i$ based on CSB(Sc)
Method:
1: Initialize CSB(Sc+$e_i$ ) = Ø.
2: For each item in ei-queue of CSB(Sc), insert its suffix sequence into CSB(Sc+$e_i$ ).
3: Return CSB(Sc+$e_i$ ).
```

FIGURE 5: The algorithm for constructing Sub-CSB.

Example For the Ini-CSB shown in Figure:4, we obtain all suffix sequences of a by following the a-queue as CSB(a), which is one of the sub-conditional sequence base of Ini-CSB. The result is shown in Figure:6 CSB(a) contains {bac:1, bcac:1, ba:1, bacc:1}. Note that bac:1 is the abbreviation of (b:1)(a:1)(c:1).

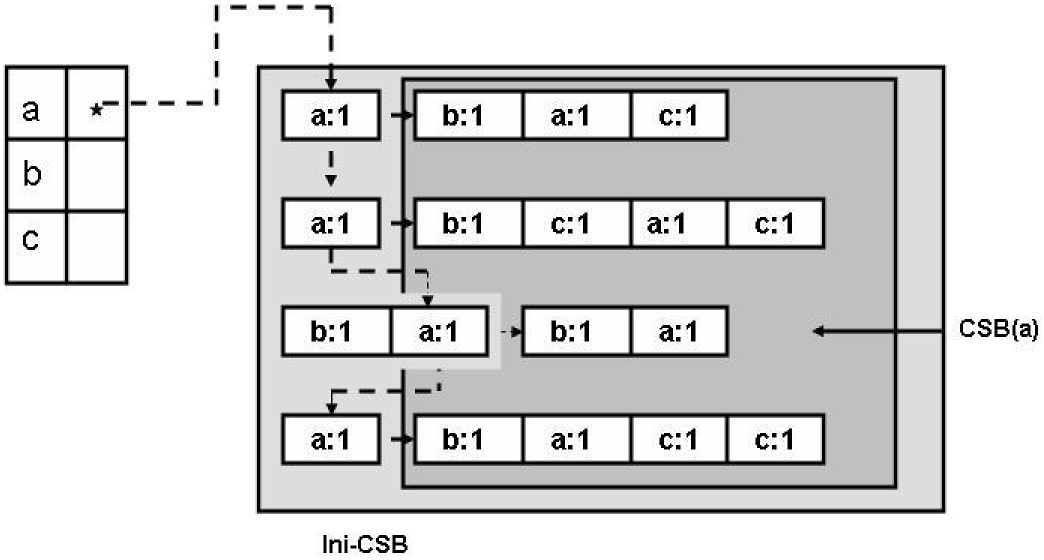


FIGURE 6: Construction of CSB(a) based on Ini-CSB.

3.2.4 Single Sequence Testing for Conditional Sequence Base

In this step, if all the sequences in CSB(Sc) can be combined into a single sequence, the mining of CSB(Sc) will be stopped. This single sequence will be used to form a part of the final periodic sequential access patterns. Otherwise, we construct Sub-CSBs for CSB(Sc) and perform recursive mining. The TestCSB algorithm for testing whether all the sequences in CSB(Sc) can be combined into a single sequence is given in Figure:7

Algorithm: TestCSB

```
Input:
1: CSB(Sc) – conditional sequence base of Sc
2: HT – Header Table of CSB(Sc)
Output:
1: test result - successful or failed flag
2: SingleSeq - single sequence of CSB(Sc)
Method:
1: Initialize SingleSeq = Ø.
2: If CSB(Sc) = Ø, return successful and SingleSeq = Ø.
3: For i = 1 to maximum length of sequences □ CSB(Sc) do
   a) If all the ith items in each sequence □ CSB(Sc) are the same event e. And if total count of
   these items ≥ MinSup X |Ini-CSB|, create a new item e with the count and insert it into
   SingleSeq.
         b) Otherwise, return failed and SingleSeq = Ø.
4: Return successful and SingleSeq
```

FIGURE 7: The algorithm for testing conditional sequence base.

Example For CSB(a) = {bac:1, bcac:1, ba:1, bacc:1}, the first item of each sequence can be combined into one item (b:4), but the second item cannot. The combination is stopped and returns the failed flag. For CSB(aa) = {c:2, cc:1}, the sequences can be combined into a single sequence c:3 and the successful flag is returned.

3.2.5 TCS-mine for Mining Periodic Sequential Access Patterns

The complete TCSM algorithm is shown in Figure:8

Algorithm: TCSM

```
Input:
1: C = [$CB_T$, $C_I$ ] – calendar-based periodic time constraint that consists of calendar
template $CB_T$ and calendar instance $C_I$
2: MinSup - support threshold
3: $WAT_{DB}$ = {$WAT_i$ |$WAT_i$ = ($t_i$, $WAS_i$ ), 1 ≤ i ≤ n} – web access transaction database,
and $WAT_i$ is a web access transaction that consists of transaction time $t_i$ and web access
sequence $WAS_i$
4: E = {$e_i$|1 ≤ i ≤ n} – all access events in $WAT_{DB}$
Output:
1: PSAP - the set of periodic sequential access patterns
Method:
1: Initialize PSAP = Ø.
2: Use ConsPreprocessing to construct Ini-CSB (CSB(Sc), Sc = Ø).
3: Use ConstructEQ to construct event queues for CSB(Sc).


4: Use TestCSB to test single sequence for CSB(Sc).
   a) If test is successful, insert all ordered combinations of items in
       frequent sequence FS = Sc+SingleSeq into PSAP.
   b) Otherwise, for each $e_j$ in Header Table of CSB(Sc), use ConstructSubCSB to construct
          CSB(Sc+$e_j$ ). Set Sc = Sc+$e_j$ and recursively mine CSB(Sc) from step3.
5: Return PSAP.
```

FIGURE 8: The algorithm for mining periodic sequential access patterns.

Length of Patterns	Periodic Sequential Access Patterns
1	a:4, b:4, c:3
2	aa:4, ab:4, ac:3, ba:4, bc:3
3	aac:3, aba:4, abc:3, bac:3
4	abac:3

TABLE 3: The periodic sequential access patterns of the sample database.

Example The complete periodic sequential access patterns with C = [(day-of-week [1, 7], hour [0, 23]), ({6, 7}, {20, 21})] and MinSup = 75% is shown in Table 3.

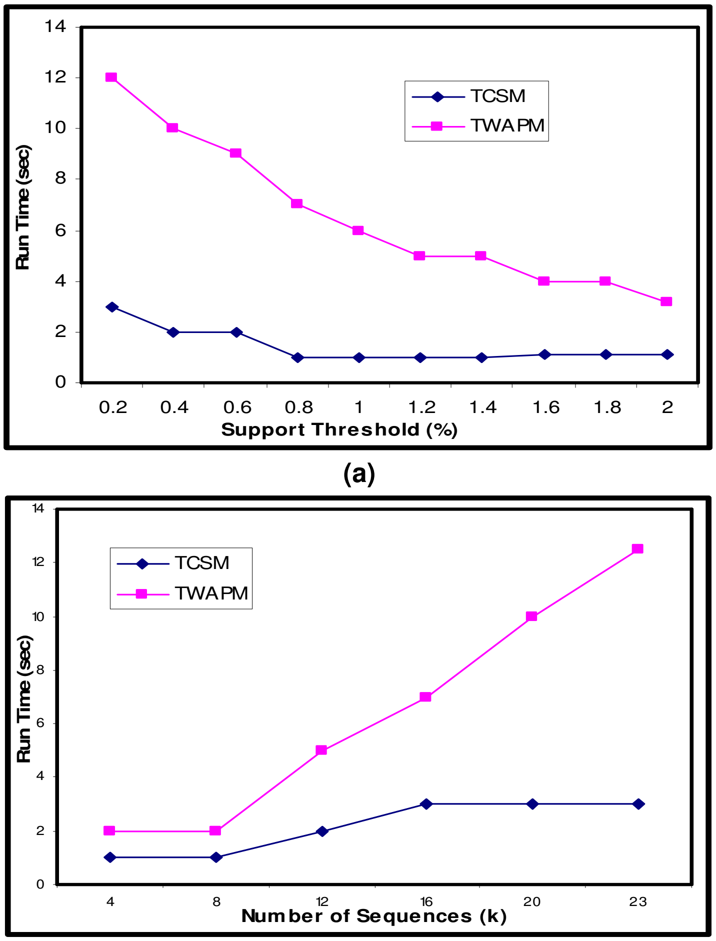
4 PERFORMANCE EVALUATION

In this section, we present the performance of TCSM and compare it with the temporal version of the Web Access Pattern mine (or TWAPM) algorithm for mining periodic sequential access patterns. Web Access Pattern mine is one of the most efficient algorithms that mine common sequential access patterns from a highly compressed data structure known as Web Access Pattern- tree. As evaluated in the performance of the Web Access Pattern mine algorithm is an order of magnitude faster than other Apriori-based algorithms. Therefore, we only compare the TCSM algorithm with the TWAPM algorithm here.

In order to deal with calendar-based periodic time constraints, the step on Constraint Preprocessing discussed in Section 3.2.1 is applied to TWAPM for extracting all the constraint-satisfied transactions from the original web access transaction database. The Web Access Pattern-tree is then constructed from the constraint-satisfied transactions, and the Web Access Pattern mine algorithm is used to mine the periodic sequential access patterns.

The two algorithms, TCSM and TWAPM, are implemented in Java. All experiments are performed on a 3.00 GHz Intel Pentium 4 PC machine with 512 MB memory, running on Microsoft Windows XP Professional. The Microsoft Anonymous Web Data is used to test the two algorithms. This dataset contains logs on which areas of www.microsoft.com each user has visited and has a total of 32,711 transactions, with each transaction containing from 1 up to 35 page references from a total of 294 pages. We set the calendar-based periodic time constraint C = [(day-of-week [1, 7], hour [0, 23]), ({1, 2, …, 5}, *)], which means every hour of every weekday. As a result, 22,717 constraint-satisfied transactions are used for the measurement.

To measure the performance, two experiments have been conducted. In the first experiment, we have measured the scalability of the two algorithms with respect to different support thresholds. This experiment uses the 22,717 constraint-satisfied web access sequences with different support thresholds (from 0.2% to 2.4%). The experimental results in Figure:9 (a) have shown that the run time of the TWAPM increases sharply, when the support threshold decreases, and the TCSM always costs less time than the TWAP-mine. In the second experiment, we have measured the scalability of the two algorithms with respect to different sizes of the constraint-satisfied web access sequences. The experiment uses a fixed support threshold (0.2%) with different databases (with sizes vary from 4,000 to 22,717 constraint-satisfied web access sequences). The experimental results in Figure:9 (b) have shown that the TCSM has better scalability than the TWAPM while the size of input database becomes larger.



(a)

(b)

FIGURE 9(a&b): Scalability with different (a) support thresholds (b) number of sequences.

5. CONCLUSIONS

In this paper, we have proposed an efficient approach, known as TCSMA for mining periodic sequential access patterns based on calendar-based periodic time constraints that can be used for describing real-life time concepts. The performance of the TCSMA has been evaluated and compared with a temporal version of the Web Access Pattern – mine algorithm. Experimental results have shown that the TCSMA performs much more efficient than the TWAPMA,especially when the support threshold becomes small and the number of web access sequences gets larger.

REFERENCES

FOR JOURNALS:

[1] Kosala R., and Blockeel H., (2000). Web Mining Research: A Survey. In *ACM SIGKDD Explorations*, Vol. 2, pp. 1-15.

[2] Ganter B., and Wille R., (1999). Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg, 1999.

[3] Cooley R., Mobasher B., and Srivastava J. (1999). Data Preparation for Mining World Wide Web Browsing Patterns. In *Journal of Knowledge and Information Systems*, Vol. 1, No. 1.

FOR CONFERENCES:

[4] Agrawal R., and Srikant R. (1995). Mining Sequential Patterns. In *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, pp. 3-14.

[5] Srikant R., and Agrawal R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements. In *Proceedings of the 5th International Conference on Extending Database Technology (EDBT)*, Avignon, France, pp. 3-17.

[6] Pei J., Han J., Mortazavi-asl B., and Zhu H. (2000). Mining Access Patterns Efficiently from Web Logs. In *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD '00)*, Kyoto, Japan, pp. 396-407.

[7] Lu H., Luo Q., Shun Y.K., (2003). Extending a Web Browser with Client-Side Mining. In *Proceedings of the 5th Asia Pacific Web Conference (APWeb)*, pp. 166-177.

[8] Ozden B., Ramaswamy S., and Silberschatz A. (1998). Cyclic Association Rules. In *Proceedings of the 14th International Conference on Data Engineering*, pp. 412-421.

[9] Ramaswamy S., Mahajan S., and Silberschatz A. (1998). On the Discovery of Interesting Patterns in Association Rules. In *Proceedings of the 24th International Conference on. on Very Large Data Bases*, New York, USA, pp. 368-379.

Key Protection for Private Computing on Public Platforms

Thomas Morris morris@ece.msstate.edu
Electrical and Computer Engineering
Mississippi State University
Mississippi State, 39762, USA

V.S.S. Nair nair@engr.smu.edu
Computer Science and Engineering
Southern Methodist University
Dallas, 75205, USA

Abstract

Private Computing on Public Platforms (PCPP) is a new technology designed to enable secure and private execution of applications on remote, potentially hostile, public platforms. PCPP uses a host assessment to validate a host's hardware and software configuration and then uses applied encryption techniques embedded in the operating system to isolate the protected application allowing its executable code, context, and data to remain unaltered, unmonitored, and unrecorded before, during, and after execution. PCPP must secure its encryption keys to ensure that the application isolation is robust and reliable. To this end we offer a protection scheme for PCPP encryption keys. We describe our PCPP key protection methodology and how it interacts with the other PCPP building blocks to isolate encryption keys even from privileged users.

Keywords*:* application isolation, encryption key protection, private computing

1 INTRODUCTION

Distributed computing technologies which enable the use of idle processors by remote users are abundant. SETI@Home [6] is a popular distributed computing application run by the University of California at Berkeley which uses remote computers to download and process data in the search for extra terrestrial intelligence. The Globus Toolkit [7] is an open source toolkit which has been used to build many working grids which are collections of computers linked to allow the sharing of computing resources across locations. Although there have been many successes in this area there has been a lack of large-scale commercial acceptance. We believe that this trend stems from a need for better application security and privacy while applications are stored and executed on remote platforms. Private Computing on Public Platforms (PCPP) was developed as a solution to this problem. PCPP enables the secure and private use of remote platforms by first performing a host assessment to validate the hardware and software configuration of the system and then using 4 active security building blocks which assure that applications executed on the remote platforms remain unaltered, unmonitored, and unrecorded before, during, and after execution.

PCPP [4] protects applications by isolating all memory and files used by the application from all other users regardless of privilege. In order to protect PCPP applications on a remote host, PCPP must have a secure and reliable means of storing encryption keys on the remote host. We have developed a new key protection methodology which isolates PCPP encryption keys from all other processes running on the same platform.. We call this new methodology PCPP key protection. PCPP key protection uses a key cache to hold all of the keys used by the other PCPP building blocks. This key cache is tagged with integrity information during the operating system's context switch routine, at the point immediately before a PCPP isolated process relinquishes the CPU to another process, and then encrypted with a master key, k_m. The master key is then securely stored until the PCPP isolated process regains the CPU, at which point, again during the operating systems context switch routine k_m is retrieved to decrypt the isolated process's key cache.

In this remainder of this paper we first offer a discussion of related works. We then provide an overview of PCPP. Finally, in the body of this paper we provide a detail description of our new PCPP key protection methodology including a description of its architecture and implementation, a description of how the key protection methodology defends against attacks, and detail analysis of the run time impact associated with using the PCPP key protection methodology.

2 RELATED WORKS

Chow et al. [14] define a white box attack context in which among other things an attacker has complete control of the execution environment. Chow then offers a methodology for creating decryption implementations in which a static decryption key is embedded in the decryption implementation such that other users or malicious programs on the same machine cannot learn the encryption key. Embedding the decryption key in the decryption implementation may stop malicious users from learning the key; however, in the white box attack context, as defined by Chow, the malicious user would have no need of the decryption key since he could just use the decryption implementation as constructed to decrypt content at will. This of course, would not provide adequate protection for PCPP isolated processes.

Perkins et al. [7] hide secret keys in the heuristic solutions of NP-hard problems. Since the heuristic solution to the NP-hard problem can be solved much faster than a brute force implementation of the same problem, the key protection problem is reduced to limiting access to the chosen heuristic and knowledge of the chosen heuristic. However, in an open system we cannot reliably limit access to the chosen heuristic and as such this solution does not fit the PCPP model. Additionally, most NP-hard problems have multiple heuristics which may provide similar solutions possibly making it easy for an attacker to learn the key without knowledge of the exact heuristic.

Trusted Computing [1][2] uses a hardware device called a Trusted Platform Module (TPM) for encryption key generation, encryption key storage, and encryption. These functions of the TPM are required to be implemented such that they are tamperproof, resistant to observation, and resistant to reverse engineering. Such a hardware based solution is ideal, however, most platforms do not have TPM's on board and we desire a software only solution.

The Linux Key Retention Service (LKRS) [4] allows applications to store and retrieve key material from key rings, which are structures which point to linked lists of keys. LKRS attaches a thread key ring pointer to each thread's task structure (a structure used by Linux to hold a thread's context). Any privileged process running may dereference another process's thread key ring and traverse a set of pointers to learn the contents of the stored key. PCPP requires isolation of keys even from privileged users. As such, LKRS is inadequate for use with PCPP as a key protection system.

3 PRIVATE COMPUTING ON PUBLIC PLATFORMS

Private Computing on Public Platforms (PCPP) is an application security technology designed to protect applications operating in public computing environments. PCPP uses a host assessment made from internal and external scans of the public platform combined with 4 active security blocks which run alongside the protected application on the public platform; the executable guard, Secure Context Switch, Secure I/O, and PCPP encryption key protection, to protect the PCPP application while it executes on the public platform. The host assessment validates the public platform by first scanning it internally and externally to collect a set of platform attributes and second classify the host as a threat or non-threat using a Bayesian classifier. The executable guard is an encrypted ELF variant which securely stores the application's binary executable on the public platform and securely loads the binary executable into memory just prior to execution. Secure context switch stops eaves droppers from accessing the PCPP protected application's volatile memory by encrypting all protected application memory context when the

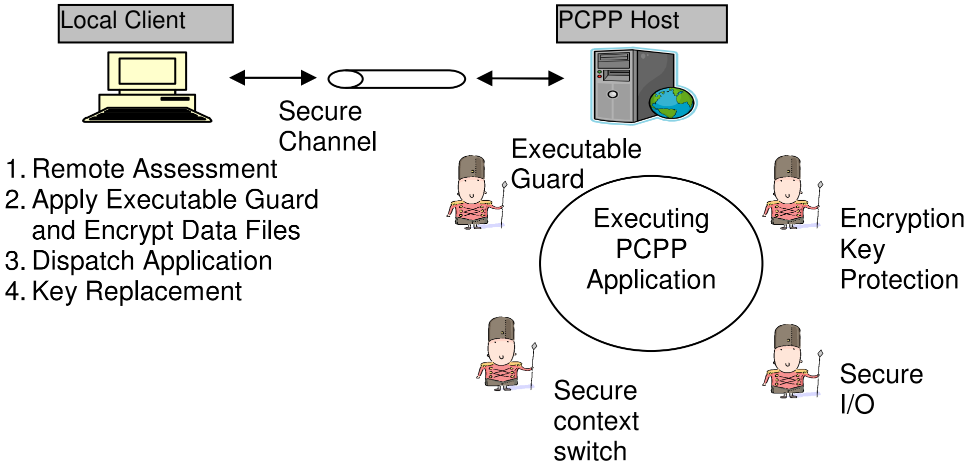


Figure 1: PCPP System Overview

application relinquishes the CPU during context switch and decrypting memory context on demand when the protected application is active. Secure I/O protects all PCPP files via decryption and encryption of all file input and output data, respectively, in the virtual file system layer of the operating system. The executable guard, secure context switch, and secure I/O all rely upon robust encryption key protection on the public platform.

4 PCPP KEY PROTECTION

PCPP key protection patches the Linux kernel scheduler's context switch routine to securely store a PCPP process's encryption keys immediately before a PCPP process relinquishes the CPU and to retrieve the stored encryption keys immediately before a PCPP process regains ownership of the CPU.

Each building block, Executable Guard (PPELF), Secure Context Switch, and Secure I/O, will have at least one encryption key and likely more than one. We use a key cache to store all of the encryption keys. The key cache uses an index to look-up and store keys as shown in Figure 2. We use a master key, k_m, to encrypt the key cache.

Using the k_m to encrypt the key cache reduces PCPP's protection burden to a single point. Building a fortress around k_m in turn protects all of the collected keys and all of the PCPP executable code and data. This fortress is a modified context switch routine which safely stores k_m during the period the PCPP protected process does not own the CPU.

The master key k_m is used every context switch by the Secure Context Switch and Secure I/O blocks. Context switches always come in pairs. If we assume the PCPP protected application is currently running and about to switch out, i.e. relinquishes control of the CPU, then during the first context switch Secure Context Switch and Secure I/O encrypt their respective data with keys stored in the key cache and then k_m is used to encrypt the key cache. The second half of the context switch pair is when the PCPP protected application switches in, i.e. regains control of the CPU, Secure Context Switch and Secure I/O use keys from the key cache to decrypt their respective data. k_m is used for exactly on context switch pair and then replaced. Each time the PCPP protected application switches out a new k_m is chosen.

Figure 3 shows a flow chart of the modified Linux context switch routine. The white boxes represent the functionality of the context switch routine before modification and the grayed boxes represent the additional functionality added to protect PCPP encryption keys. The unaltered context switch routine performs two simple steps. First, a pointer to the current processes memory map is changed to point to the memory map of the incoming process. Second, a small block of assembly code is called to swap out the CPU registers from the old process replacing them with the register values belonging to the incoming process. The PCPP updates to the context switch routine introduce two new paths through the context switch routine, one to handle outgoing PCPP processes and one to handle incoming PCPP processes.

The context switch routine is passed two pointers, *prev* (previous task) and *next (next task)*, when called, each is a pointer to a Linux task structure. The *prev* task structure holds the context belonging to the process relinquishing the CPU and the *next* task structure holds the context of the process which is gaining control of the CPU. We modified the PCPP task structure to contain a variable called *pcpp*. The *pcpp* variable is a Boolean which defaults to FALSE. When a PCPP process is launched the executable guard sets the *pcpp* variable to TRUE.

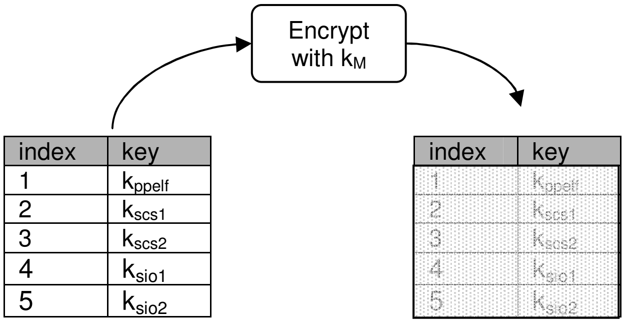
When *prev* is a PCPP process the modified context switch routine first encrypts any memory pages belonging to *prev* which are not currently in an encrypted state (this step is actually part of the Secure Context Switch building blocks and which is described in greater detail in **Error! Reference source not found.**). The encryption keys used to encrypt the context are held in a PCPP key cache which is attached to *prev's* task structure. When a PCPP process owns the CPU the key cache is in a decrypted state. It is encrypted as the PCPP is context switched out. Before encrypting the key cache we first add an integrity hash value to the key cache. The key cache is then encrypted using a master key, k_m. Finally, k_m is stored to await retrieval when the PCPP process regains control of the CPU.

When *next* is a PCPP process, the right hand path of the context switch routine is taken. First, k_m is retrieved from storage. Next, the key cache is decrypted. After the key cache is decrypted the integrity hash value, stored in the key cache when this PCPP process last relinquished the CPU, is validated. If the integrity hash value is incorrect a PCPP process shutdown commences. This shut down erases all PCPP files, overwrites all PCPP memory pages, and then kills the PCPP process. If the integrity hash value is correct the context switch will continue.

It is possible for both the previous thread and the next thread to be PCPP threads. In this case both branches of PCPP code in the context switch routine will be run, first the *prev.pcpp* branch, then the *next.pcpp* branch. In this case, the two PCPP tasks would each have separate key caches and master keys. As such encrypting *prev's* key cache and master key would not affect the retrieval of *next's* master key or decryption of *next's* key cache.

Table 1: PCPP Key Cache Integrity Hash Contents

Index	Item	Description
x_0	pcppcs_start	Physical address of the first instruction in PCPP context switch code
x_1	pcppcs_end	Physical address of the last instruction in PCPP context switch code



index	key
1	k_{ppelf}
2	k_{scs1}
3	k_{scs2}
4	k_{sio1}
5	k_{sio2}

index	key
1	k_{ppelf}
2	k_{scs1}
3	k_{scs2}
4	k_{sio1}
5	k_{sio2}

Figure 2: Encryption Key Cache

x_2	pcppcshash	Hash of the PCPP context switch code
x_3	keycache_loc	physical address of the PCPP key cache
x_4	tgid	process ID of the protected application
x_5	instrptr	physical address of random location in PCPP process's intructions
x_6	instrhash	hash of random 32 bytes from instruction memory

The integrity hash is a single value generated from the items listed in Table 1. Starting with the first value, x_0, we xor each value with the previous hash and then hash that intermediate result. The resulting hash will change if any of the items from Table 1 changes. Equation 1 illustrates the process for creating the integrity hash mathematically.

$$i_{n+1} = h(x_n \oplus i_n) \tag{1}$$

First, we store the starting and ending physical addresses of the PCPP context switch code (the grey boxes from Figure 3). These are used to confirm, after decrypting the key cache, when the PCPP process regains control of the CPU, that the PCPP context switch code has not moved. Since, this code is in the kernel address space it cannot be paged out and moved, it should stay in the same location permanently. The next integrity item is a hash of the PCPP context switch code. This item is used to confirm that the PCPP context switch code has not changed since the PCPP process last relinquished the CPU. The next item is physical address of the key cache itself. The key cache physical address is used to confirm that the key cache has not moved since the PCPP process last relinquished the CPU. The next item is the process ID of the protected process. The process ID is used to confirm that the process decrypting the key cache matches the process which previously encrypted it. The next item is the address of a random location in the PCPP process's instruction code. This address is the start location for hash of 32 random bytes from the PCPP process's instruction code. This hash is used as a spot check to confirm the protected process's instructions were not altered. If the integrity hash value found in the key cache after decryption does not the match the hash computed at the time of decryption the integrity

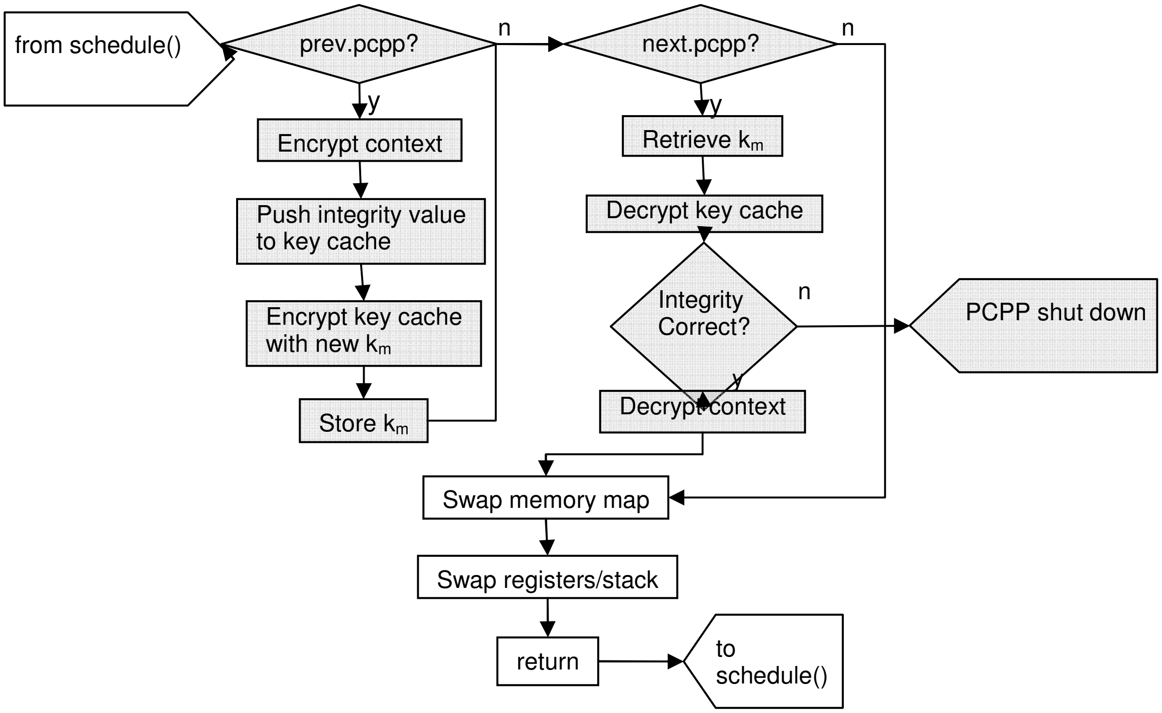


Figure 3: PCPP Context Switch Flowchart

check fails and the PCPP process is safely killed.

We developed two similar methods for storing k_m. Neither stores k_m directly, both store k_m indirectly via knowledge of the 1st and last members of a hash chain. k_m is the $(n\text{-}1)^{th}$ member of a n-length hash chain such as that shown in equation 1, where n is the number of successive hashes computed to find $h_n(h_0)$. We store h_0 and h_n as indicators for how to retrieve k_m. Retrieving k_m requires computing the hash chain from the starting value h_0 until h_n is reached and then setting $k_m = h_{n-1}$.

$$h_n(h_0) = h(h(...h(h_0))) \qquad 2$$

4.1 HMAC BASED HASH CHAINS

Protecting the master key is a difficult problem. In order to retrieve the key after storage we must also store some information on how to retrieve it. However, when the PCPP process relinquishes the CPU a foe may access this same information and couple that with knowledge of our hash chain implementation to recreate the hash chain. It is not possible to store a secret we can use which absolutely cannot be used by a foe. The SHA1/MD5 method stores h_0, h_n, and the hash type, which are then used to retrieve the key. We have shown in section via our LKRS exploit that privileged users can easily access the memory contents of other processes. This makes storing h_0, h_n, and the hash type in the clear dangerous. However, to hide these or encrypt them we would need a new secret and that secret would need to be stored. There is always a last secret which must be stored where a foe may conceivably find it. We developed to our HMAC [13] based hash chains to make it more difficult, though, we cannot make it impossible.

To make retrieving the master key more difficult for a foe we desire several properties. First, we prefer to choose from many hash algorithms. Second, we prefer to construct the hash implementations in such a way that any foe finds it much easier to try to use our hash implementations rather than build his own implementations and simply use his to create a copy of our hash chains.

Since, HMAC is a keyed hash we can use different keys to generate separate implementations. In fact, we build many separate HMAC functions with predefined keys embedded in the HMAC implementation executable code. Since, the HMAC algorithm implementations are distinct and built with different keys; they meet the requirement of many different hash choices.

The keys used for each HMAC implementation are defined as constants in their respective implementations and copied into memory as the first step when the HMAC implementation is called. Because the keys are constants and too large to copy with a single *mov* assembly instruction they are stored in parts in the executable code and copied into registers when the HMAC implementation is called. After the HMAC is computed we overwrite these registers with zero. Storing the HMAC keys in parts as constants in the executable code, rather than in the processes task structure or some other more accessible location serves to make it more difficult for a foe to steal the HMAC keys, though definitely not impossible.

Ideally, the HMAC implementations could be further obfuscated to make even more difficult for a foe to find the keys in the HMAC executable code. Currently, the keys are stored as 4, 32-bit constants and loaded into registers with 4 *mov* calls. Finding, them would be somewhat difficult, but certainly not impossible. If the code were obfuscated to diffuse the keys further into the HMAC implementation it would become more difficult for a foe to find the HMAC keys. This would serve to encourage a foe to attempt to run our HMAC implementation rather than try to steal the HMAC keys and run offline.

It is important that the HMAC implementations only be used for one PCPP application run. Re-use of the same HMAC implementations with the same embedded keys would give foes extra time to learn the HMAC keys. The HMAC implementations can be derived on the PCPP local client and sent to the PCPP remote host during application launch or they can be created on the remote host loaded into memory when the PCPP application launches.

Equation 3 describes the ordinary HMAC. Ordinary HMAC takes a single key and xor's that key with separate ipad and opad values.

$$hmac(x) = h(k \oplus opad, h(m, k \oplus ipad)) \quad 3$$

We found it simpler to dimply use two separate keys rather than create one key and then XOR twice for use in the HMAC. Our HMAC is shown in equation 4.

$$hmac(x) = h(k_1, h(m, k_2)) \quad 4$$

We first choose h_0 using a random number generator. We also use the random number generator to choose the length of the hash chain and to choose the underlying hash algorithm used to compute the HMAC this chain is derived from. The length of the hash chain must be greater than 16 and less than 128 links. Our current implementation chooses between MD5 and SHA1, though other hash algorithms would be acceptable as long as they meet HMAC requirements.

We store h_0, h_n, and the hash type in the PCPP processes task structure. The hash type now indicates both the underlying hash algorithm and which HMAC implementation to use. The h_0, h_n, and the hash type values are still stored in the clear when the PCPP process relinquishes control of the CPU. As mentioned above, k_1 and k_2 are embedded in the HMAC executable code. When the PCPP task regains control of the CPU h_0, h_n, and the hash type are used to call the appropriate HMAC implementation to recreate the hash chain and derive k_m.

4.2 TRANSFERRING KEYS FROM THE LOCAL CLIENT TO THE REMOTE HOST

In addition to safe storage of encryption keys on the remote host, the initial master key and key cache must be securely transmitted to the remote host from the local client.

The PCPP host must run a server which receives the PCPP application executable, any files sent with the executable, a hash chain, and an encrypted key cache from the local client. Once the server receives the complete launch packet it launches the application on the remote host. This PCPP server must use SSL to encrypt the connection between the local client and the remote host. It must also be protected by all the PCPP building blocks like any other PCPP application to ensure the PCPP servers executable and data remain unaltered, unmonitored, and unrecorded.

Table 2: Initial Key Transfer

Index	Who	Step
1	Remote Host	Send HMAC keys over SSL secure channel
2	Local Client	Create key cache: stores initial encryption keys for PPELF and any files sent with the executable
3	Local Client	Create initial master key (k_m), and hash chain
4	Local Client	Encrypt key cache
5	Local Client	Send key cache and hash chain to remote host
6	Remote Host	Store key cache and hash chain to server PCPP structure
7	Remote Host	Launch application
8	Remote Host	New process inherits key cache and hash chain from parent

Table 18 shows the steps required to securely send the initial master key and hash chain from the local client to the remote host. The executable and any files to be sent with the executable must be encrypted before being sent to the remote host. This process may take place immediately before sending the executable and files to the remote host or it may happen sometime in advance. Either way the encryption keys used for this process must be sent to the remote host with the executable and any other files. These encryption keys are placed in a key cache as described above and the key cache is encrypted with the initial master key k_m. Immediately, before this encryption step a hash chain is built which is used both to derive and store k_m. The encrypted key cache and the hash chain are then sent to the remote host. All communication between the local client and remote host is encrypted to prevent third party eaves dropping. Once on the remote host the PCPP server stores in the hash chain and key cache in its own PCPP structure. The PCPP server then uses exec to launch the PCPP application. When the new PCPP process is launched it inherits a copy of its parent's task structure. If the parent is a PCPP process, which in this case it is, the parent's task structure will contain a PCPP structure. The parent PCPP structure contains a pointer to second PCPP structure, which is intended for copying to a child during launch. When a PCPP process's task structure is copied the copying code first searches for a child PCPP structure. If a child PCPP structure is available this structure is copied and used as the new process's PCPP structure. If the child PCPP structure does not exist, the parent's PCPP structure is used. For the case when the PCPP server is launching PCPP applications there will always be a child PCPP structure. When PCPP applications launch children they likely will not have PCPP structures and will therefore inherit the PCPP structure, complete with key cache and key chains from the parent.

The first item in Table 2 is specific to the HMAC case. Here, the PCPP server must send a pair of HMAC keys to the local client which can be used to create the hash chain.

4.3 DEFENSE AGAINST ATTACKS

We foresee three types of attacks against the PCPP key protection system. First, an attacker may attempt to jump midway into the key protection code in an attempt to bypass certain checks. Second, an attacker may attempt to learn the master key, k_m, by implementing his own key retrieval code or copying the PCPP key retrieval code to a separate location and modifying it. Finally, an attacker may attempt to copy a PCPP process's Linux task structure, the PCPP process's PCPP structure, and all or part of its encrypted memory contents to build separate task structure which is then placed on the ready to run queue for decryption by the context switch routine. In the remainder of this section we describe how the PCPP key protection system defends against these attacks.

In the first attack case a foe may attempt to jump to an intermediate point in the PCPP context switch code expecting to find a function return which will pop the foe's return address from the stack and return control to the attacking program. For example an attacker may wish to jump to the master key retrieval code expecting the master key to retrieve and then expecting a conveniently placed return to send control back to the attacking program. We stop such an attack by compiling all of the PCPP context switch code, shown in Figure 3, as inline code. This means all function calls are replaced by the compiler with a unique copy of the function placed in line with the calling code where the function call was previously. This stops the PCPP context switch code from calling functions using the *call* assembly instruction and then using a *return* mnemonic to jump back to the calling location. By doing this we avoid attackers jumping directly to the first instruction of a PCPP function and then using our own return call to jump out of the routine. By in-lining all PCPP context switch code the first encountered return instruction occurs at the end of the entire context switch routine. As such anytime an attacker jumps to any instruction in the PCPP context switch routine it must run through to the end of the PCPP context switch routine. Jumping into the context switch code without properly setting pointers to a previous task and a next task will cause the operating system to lock up, ceasing all activity on the machine until a reboot is performed. If an attacker manages to properly create pointers to a previous and next task the result would still almost definitely be a locked up system. In-lining stops foe's from jumping into the context switch routine expecting to return out before a context switch.

The second attack scenario involves an attacker either copying the PCPP master key retrieval code to a separate location or using his own hash implementation, with a pilfered hash chain initialization value, end value, and hash type, to retrieve the master key. Copying the key retrieval code and executing it in another location is difficult but possible. Since the code is in-lined there will be no symbols in the instruction code pointing to the starting address of the key retrieval code. Also, there will be no obvious way to know where the key retrieval code ends. If a foe finds the start and end of the key retrieval code he will then need to copy it and then add code to protect the integrity of his own codes registers by identifying all registers used in the copied code and pushing these to the stack at the beginning of the copied code and popping them from the stack when returning to his own code. If all of this is done the copied code could be used.

Instead of copying the key retrieval code to separate location an attacker may choose to only copy the hash chain initialization value, the hash chain end value, and the hash type value from the victim process's PCPP structure. We make this very difficult by using HMAC hash chains with HMAC keys embedded in the HMAC implementations as constants and by using an HMAC algorithm based upon two keys rather than the standard HMAC which is based upon just one. To successfully use his own HMAC code first the attacker would need to implement an HMAC which mimics the PCPP HMAC behavior. This would not be overly difficult since the code could be copied from the PCPP installation. Next, the foe would need to steal the HMAC keys. Since these are embedded in the HMAC executable code which is in-lined in the rest of the PCPP context switch code these key may be difficult to find. Furthermore, if the HMAC implementations were obfuscated to hide or diffuse the keys in the HMAC implementation finding these keys would be all the more difficult.

Both copying the key retrieval code and executing it elsewhere and copying just the inputs to the key retrieval code and deriving the master key from a separate hash implementation are possible attacks. However, both are considerably difficult. If an attacker does manage to retrieve k_m he may then proceed to decrypt the key cache. In this case he will learn the values of the protected keys in the key cache. To limit the damage in case this does happen we choose a new encryption key each time an isolated page is encrypted in the demand encryption/decryption algorithm. By doing this we limit the useful life of keys stolen by an attacker. We also change k_m each time a PCPP context switch relinquishes the CPU.

The last attack vector involves an attacker copying all or part of a PCPP process's context to a separate process's memory space and then attempting to use the PCPP context switch code to decrypt the copied PCPP memory but still switch CPU control to a program chosen by the attacker. If an attacker attempts to build a Linux task structure which points to a set of PCPP data pages but replaces the instruction memory pages with a separate program the context switch code will decrypt the PCPP data pages as the attacker desires. However, it will also attempt to decrypt the replaced instructions with a key from the key cache. Two things can happen to the instructions at this point. If the instructions were not encrypted, or encrypted with a different key than the one found in the key cache, the instructions would be mangled by the decryption step and consequently the integrity check would fail causing the PCPP shutdown process to run. If the attacker managed to encrypt his replacement program pages with the same key used for the actual PCPP application then the replacement program would decrypt correctly. However, the integrity check would still fail because the hash of the random 32 bytes of instruction code will not match the hash pushed into the key cache when the PCPP application most recently relinquished the CPU.

4.4 PCPP KEY PROTECTION PERFORMANCE OVERHEAD

There is a run time performance overhead associated with using the PCPP Key protection system. This overhead is limited to increasing the time required to complete a context switch. There is no performance degradation outside the context switch routine. We measure overhead for two configurations of the PCPP key protection system. First, we measured the overhead for a system which stores the master key with SHA1 and MD5 hash chains. Second, we measured the overhead for a system which stored the master key with HMAC based hash chains.

Figure 4 shows the context switch run time overhead for 4 PCPP key protection configurations. For all of the measurements we built a small routine which solely performed the key protection steps. The steps when context switching out include: create new master key, create integrity value, add integrity values to key cache, and encrypt the key cache. The steps when context switching in include: retrieve the master key, decrypt key cache, calculate integrity values, and validate the integrity values in the key cache. To accurately measure the time to perform these steps we built a loop which perform each step in sequence. We then timed this loop while it executed 100,000 times. The time for one loop iteration was the total measured time over 100,000. The numbers are the time for 1 context switch. Since the key retrieval operations and key storage operations are close to mirrors of one another we divide the time for one iteration by 2 to get the overhead for 1 context switch.

The measurements for Figure 4 were performed on a workstation running Linux kernel 2.6.2.20 [9] with an AMD CPU clocked at 3 GHz.

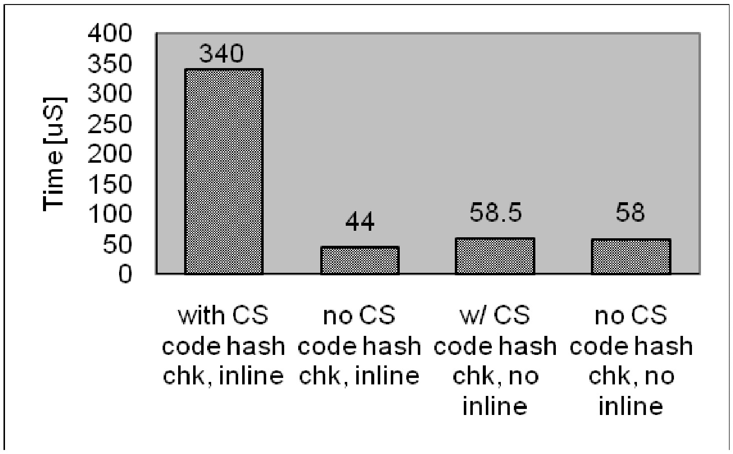


Figure 4: Context Switch Overhead from Various Key Protection Configurations

The first pair of results in left most position of Figure 4 show the run time overhead when all PCPP context switch code is compiled with the inline attribute set and with the use of a hash of all of the PCPP context switch code in the key cache to confirm the context switch code did not change since key cache encryption. The overhead for this configuration is large for both the MD5/SHA1 version and the HMAC version. This overhead is dominated by the time to compute the hash of the in-lined PCPP context switch code. Analysis of this code shows that the in-lined version of the measure code listed above was over 54K bytes. The second set of bars in Figure 4 show the impact of removing the hash of the PCPP context switch code. With this hash removed the overhead drops from 340uS to 44uS.

The next two data sets in Figure 4 show the impact of compiling the PCPP context switch code without the in-line attribute. For both cases the context switch overhead is similar to, but slightly higher than, the overhead for the case which used in-lined code but skipped the hash check of the context switch code. We expect these versions to be slightly slower than the in-lined version since the overhead of dealing with function calls is added into these results.

We conclude from this that in-lining the context switch code causes the code to grow significantly in size. This by itself would generally be acceptable since most hosts would have plenty of memory to accommodate a larger context switch routine. However, the larger in-lined code size does lead to slow hash times for validating the PCPP context switch code. We have explained the necessity of this validation step above and feel it is a required step to ensure the security of the PCPP key protection system.

We derived a set of equations to predict overall performance impact of using the PCPP key protection system.

$$t_{kprot} = e\mathrm{T} + \eta t_{out} + \eta t_{in} \qquad 5$$

Equation 5 provides an approximation of the run time required to run an application with Secure Context Switch enabled. Equation 5 starts by representing the amount of time a program needs to run in terms of context switch time slices, or epochs, *e*, times the period of 1 epoch, T. As such *e*T is the amount of run-time the program would require if it were running without any interruption, i.e. in an environment free of context switches. We represent this basic run-time as *e*T because *e* provides a minimum number of T length time slices required to run the program. All other terms in equation 5 are overhead associated with context switching.

The second term of equation 5, ηt_{out}, adds the context switch time for η context switch out(s), i.e. relinquishing the CPU. The third term of equation 5, ηt_{in}, adds the context switch time for η context switch in(s), i.e. regaining control of the CPU. In the second and third terms η is the adjusted number of times slices used by the application. We adjust η to account for the time needed for an ordinary context switch, extra time during context switch out to generate a new master key hash chain, calculate and store integrity information, and encrypt the key cache, and extra time during context switch in to retrieve the master key, calculate and check integrity information, and decrypt the key cache.

Table 3: Key Protection Overhead Variable Definitions

Variable	Description
t_{kprot}	Run time with PCPP key protection
t_{norm}	run-time without PCPP key protection
e	number of complete times slices needed to complete execution
T	period of one time slice
η	adjusted number of times slices after adding PCPP key protection overhead
t_{cs}	Context switch time without PCPP key protection
t_{in}	time for protected application to context switch in
t_{out}	time for protected application to context switch out
t_{key}	time to retrieve/store master key, encrypt/decrypt key cache, and perform integrity checks

$$t_{out} = t_{in} = t_{cs} + t_{key} \qquad 6$$

Equation 6 shows the relationship between t_{in} and t_{out}. For our PCPP key protection implementation t_{in} and t_{out} are set to equal one another because the code for context switching in mirrors the code for context switching out.

$$\eta = \left(\frac{2et_{cs} + 2et_{key}}{\mathrm{T}} \right) + e \qquad 7$$

Equation 7 estimates the adjusted number of time slices a PCPP key protected process will require. Equation 7 adds the minimum number of uninterrupted time slices, *e*, to the number of extra time slices resulting from the extra time required for basic context switching and extra time required for encrypting and decrypting protected pages during the context switch. The first term of equation 7 calculates the extra times slices required for context switching by first summing the time for 2 ordinary context switches and 2 master key retrievals/stores and then dividing by the period of one time slice, T. We use 2 context switch times and 2 master key retrieval/store times because context switches come in pairs, one at the beginning of the time slice and one at the end of the time slice.

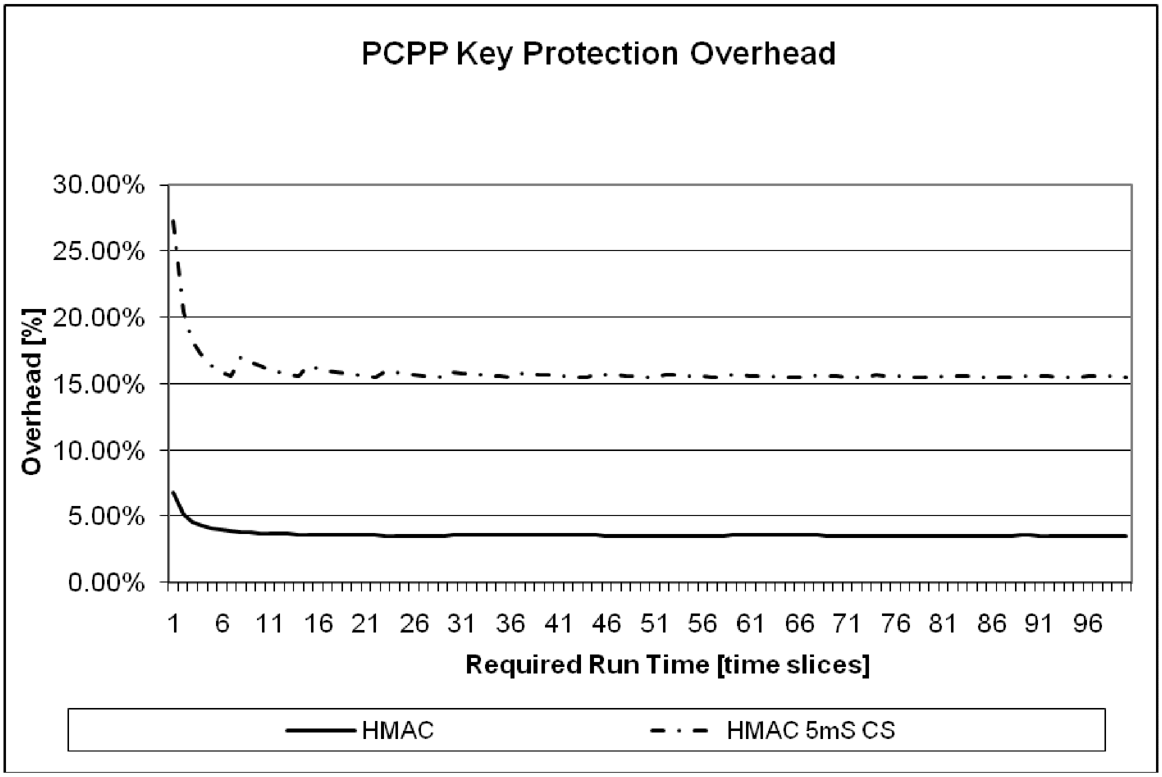
$$t_{norm} = e\mathrm{T} + \eta t_{out} + \eta t_{in} \qquad 8$$

Equation 8, which computes t_{norm}, looks just like equation 5. The only difference is the definitions of η, for the number of adjusted context switches, and the definitions for t_{in} and t_{out}

$$t_{in} = t_{out} = t_{cs} \qquad 9$$

Equation 9 defines t_{in} and t_{out}. Since there is no master key retrieval/storage, no integrity checks, and no key cache encryption/decryption required, the t_{key} term is removed relative to equation 6.

Figure 5: Predicted PCPP Key Protection Overhead



$$\eta = \left(\frac{2t_{cs}}{\mathrm{T}} \right) + e \qquad 10$$

Equation 10 shows the definition of η for the non-PCPP key protection case. Since, without PCPP key protection there is no master key retrieval/storage, no integrity checks, and no key cache encryption/decryption this term is removed relative to equation 7.

$$kprot\_overhead = \frac{t_{kprot}}{t_{norm}} \qquad 11$$

Equation 11 defines the overhead for PCPP key protected applications when compared to the same application running without PCPP key protection.

PCPP key protected applications when compared to the same application running without PCPP key protection.
We used equations 5-11 to plot the predicted overhead associated with using PCPP key protection. Figure 5 shows the predicted run time overhead for the HMAC implementations with in-lined code and all integrity checks in place. We plotted two curves. One set based upon the assumption of a 20mS context switch period, meaning our application runs in 20mS uninterrupted segments. The second set of curves show the predicted overhead for a 5mS context switch period. The chart shows that as our application is allowed to run in longer uninterrupted segments the overhead from context switching decreases. For the 5mS case we see overhead ranging from 10-15%. For the 20mS case we see overhead ranging from 3-5%. While this overhead is significant, we believe it is acceptable for the improved key protection gained by using the PCPP key protection system.

When running our key protection code with actual programs we see overheads which vary widely from job to job but tend to stay in the 2-15% range. This matches the ranges seen in Figure 5.

The overwhelming majority of the key protection overhead comes from hashing the inline code section which we use to ensure the integrity of the key protection code and our PCPP context switch code. Improving the run time performance of the key protection code can be done by reducing the size of the hashed inline code section. One might choose to hash less code, thereby choosing to guard the integrity of less code.

5 CONCLUSIONS

In this paper we presented a modified Linux context switch routine which encrypts a PCPP key cache and with a master key, k_m, when a PCPP process relinquishes control of the CPU. After encryption of the key cache k_m is securely stored. When the PCPP process regains control of the CPU, k_m is retrieved and then the PCPP key cache is decrypted. Before allowing a PCPP process to resume ownership of the CPU integrity information stored in the key cache when the PCPP process relinquished ownership of the CPU is validated. k_m is not stored as plaintext on the host platform, rather k_m is the n-1th member of a hash chain in which the root of the hash chain and the nth element of the hash chain are stored. The hash algorithm is chosen on the local client randomly and sent to the host platform during application launch.

While our PCPP key protection system is still vulnerable to other privileged processes stealing k_m when the PCPP process is not running this is made computationally difficult throught the use of HMAC based hash chains and the integrity information validation. Also, coupling the PCPP key protection mechanism with our Secure Context Switch technology limits the useful life k_m. We believe our PCPP key protection methodology is a considerable improvement over the Linux Key Retention Service. Overall, we believe that our PCPP key protection system is a significant improvement over other software based key protection systems.

REFERENCES

[1] Marchesini, J., Smith, S., Wild, O., MacDonald, R., Experimenting with TCPA/TCG Hardware, Or: How I Learned to Stop Worrying and Love The Bear, Dartmouth Computer Science Technical Report TR2003-476, ftp://ftp.cs.dartmouth.edu/TR/TR2003-476.pdf

[2] Trusted Computing Group Fact Sheet, https://www.trustedcomputinggroup.org/about/FACTSHEET_revised_may_07.pdf

[3] Felten, E.W., Understanding Trusted Computing: Will its benefits outweigh its drawbacks?, IEEE Security and Privacy Magazine, Volume 1, Issue 3, May-June, 2003

[4] Morris, T. Nair, V.S.S. Private Computing on Public Platforms: Portable Application Security. Submitted to Wiley InterScience Journal of Wireless Communications and Mobile Computing. (to appear)

[5] Kumar A., Chopdekar S., Getting Started with the Linux key retention service, http://www.ibm.com/developerworks/linux/library/l-key-retention.html

[6] Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M., and Werthimer, D. 2002. SETI@home: an experiment in public-resource computing. *Communications of the ACM* 45, 11 (Nov. 2002), 56-61.

[7] Foster, I. Globus Toolkit Version 4: Software for Service-Oriented Systems**.** IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2005.

[8] Perkins, G., Bhattacharya, P., An Encryption Scheme for Limited k-time Access to Digital Media, IEEE Transactions on Consumer Electronics, Volume: 49, Issue: 1, Feb. 2003

[9] The Linux Kernel Archives, http://www.kernel.org/

[10] Barak, B. and Halevi, S. 2005. A model and architecture for pseudo-random generation with applications to /dev/random. In *Proceedings of the 12th ACM Conference on Computer and Communications Security* (Alexandria, VA, USA, November 07 - 11, 2005). CCS '05. ACM, New York, NY, 203-212.

[11] D. Eastlake and P. Jones. RFC 3174. US Secure Hash Algorithm 1 (SHA1). http://www.faqs.org/rfcs/rfc3174.html

[12] R. Rivest. RFC 1321. The MD5 Message-Digest Algorithm. http://www.faqs.org/rfcs/rfc1321.html

[13] H. Krawczyk, M. Bellare, and R. Canetti. RFC 2104. HMAC: Keyed-Hashing for Message Authentication. http://www.faqs.org/rfcs/rfc2104.html

[14] Chow, S. Eisen, P. Johnson, H. Van Oorschot, P. A White-Box DES Implementation for DRM Applications. Digital Rights Management. Springer-Verlag LNCS 2696, pp 1-15, 2002.

A Novel Luby-Rackoff Based Cipher in a New Feistel-Network Based LPRKES for Smart Cards

Ehab Mahmoud Mohamed ehab@mobcom.is.kyushu-u.ac.jp
Faculty of Engineering/
Advanced Information Technology Dept/
Wireless Communication Section/Kyushu University
Motooka 744, Nishi-ku, Fukuoka-city 819-0395, Japan
Phone +81-92-802-3573, Fax +81-92-802-3572,

Yassin Mahmoud Yassin Hasan ymyhasan@aun.edu.eg
Faculty of Engineering /Electrical Engineering Dept/
Electronics and Communication Section
Assuit University
Assuit, Egypt.

Hiroshi Furukawa furuhiro@is.kyushu-u.ac.jp
Faculty of Engineering/
Advanced Information Technology Dept/
Wireless Communication Section/Kyushu University
Motooka 744, Nishi-ku, Fukuoka-city 819-0395, Japan
Phone +81-92-802-3573, Fax +81-92-802-3572,

Abstract

The RKES (Remotely Keyed Encryption Schemes) are greatly useful in solving the vital problem of how to do bulk encryption and decryption for high-bandwidth applications (like multimedia and video encryption) in a way that takes advantage of both the superior power of the host and the superior security of the smart card. In this way, we propose a novel length preserving (LP) RKES by using a proposed general view of Feistel-Network (FN) in which we use only two rounds in an efficient way. The proposed LPRKES needs a strong pseudorandom permutation (PRP) as its basic building block, so we introduce a new symmetric-key block cipher, with variable block and key lengths, referred to as NLMSFC (Nonlinear Matrix Structure Based Feistel Cipher), appropriate for hardware and software implementations. NLMSFC is a 3-round Luby-Rackoff construction. In this structure, robust pseudorandom functions (PF) are used to obtain a pseudorandom permutation (PRP). NLMSFC makes use of a novel PR keyed-subfunction in a matrix like structure. Extensive statistical tests are conducted upon NLMSFC and its round function in order to demonstrate their competitive diffusion, confusion and pseudorandomness characteristics. In addition NLMSFC is provably secure. At the end of this paper, we show how we can apply NLMSFC as a strong PRP in the suggested LPKES to be used for cryptographic smart cards.

Keywords: pseudorandom function (PF), pseudorandom permutation (PRP), Luby-Rackoff ciphers, Feistel Network (FN), LPRKES.

1. INTRODUCTION

Smart cards provide an effective tool for portable safe hardware storage of secret keys critically needed in many recent multimedia applications such as real time access control, software license management, e-technology, e-commerce and e-services [1]. Smart cards are mainly reliable because of their distinctive features of tamper-resistant packaging, loose coupling to the host and low cost [2]. However, with their computationally limited resources, smart cards cannot process large data blocks as fast as the host may need.

The Remotely Keyed Encryption Protocol RKEP), first introduced by Blaze, addressed how to do bulk encryption/decryption taking advantage of both the superior computational power, speed and resources of the (high bandwidth) host (trusted with plaintexts/ciphertexts) and the superior security of the slow (low bandwidth) smart-card (trusted with the key) [2]. Although of the interesting approach of Blaze, it suffers from some drawbacks. Its drawbacks basically result from the low security of the protocol. Lucks gave three attacks on the blaze's RKEP, namely a chosen plaintext attack, a two sided attack and a forgery attack (working on the decrypt only smart-card) [3]. In addition, Lucks specified three conditions, that Blaze's RKEP does not satisfy any of them, to make a secure RKE scheme (RKES). Moreover, Lucks suggested the RaMaRK "Random Mapping based RKES" which is based on the Luby-Rackoff construction. Although RaMaRK is based upon Lucks' criteria, a critical weakness was found in RaMaRK [4]. Consequently, Blaze, Feigenbaum and Naor suggested two general RKESs, classified based on the relative length of the ciphertext compared to the plaintext as: a length-preserving (LP) RKES and a length increasing (LI) RKES (with self validation), referred to as BFN-LPRKES and BFN-LIRKES, respectively [4]. To achieve self-validation in the BFN-LIRKES, a signature of the whole ciphertext is appended to the output ciphertext which cannot be computed by an adversary without running the encryption protocol. So any adversary cannot forge the scheme.

In this research, both the fact that, in order to produce a 2n-bit PRP (with entropy of 2n) from n-bit PRF (with entropy of n), it theoretically needs at least two rounds of n-bit PRFs and the fact that the main reason recalling for excess rounds in the Luby-Rackoff construction (and FN ciphers in general) is the rounds joining XOR function, motivated us to construct such a 2-round (only) network excluding the XOR and use a PRP instead. So, in this paper, we develop a new LPRKES employing only a 2-round network based on a general view of an unbalanced Luby-Rackoff construction. The proposed LPRKES is forgery secure, inversion secure and strong pseudorandom. The proposed LPRKES is more secure than the Blaze's RKEP and RaMaRK, more efficient than RaMaRK and the BFN-LPRKES from the card computations and key storage point of views, and requires less number of interactions between the host and the card than the BFN-LPRKES. In addition, the authors proposed an efficient and secure LIRKES [5].

Because of the requirement for a strong PRP in the proposed LPRKES, we introduce NLMSFC: Nonlinear Matrix Structure based Feistel Cipher as variable block-size symmetric-key block cipher. Block cipher is a PRP that maps a block of bits called plaintext into another block called ciphertext using the key bits. Pseudorandomness implies being not distinguishable form truly random permutation (TRP). In a well designed block cipher, a plaintext bit change should change each bit of the output ciphertext with a probability of 0.5. Also, there should be no plaintext/ciphertext-to-ciphertext correlations. Thus, secure block ciphers should essentially exhibit high degree of pseudorandomness, diffusion, and confusion [6]. In addition, a block cipher is most practically qualified as secure if it has survived after being extensively exposed to proficient cryptanalysis. The structure of a block cipher may be a substitution-permutation network (SPN) or Fesitel network (FN). The Advanced Encryption Standard AES-Rijndael is currently the most famous SPN cipher [7]. Alternatively, the FN structure, which is a universal method for converting a round function into a permutation, is adopted in several ciphers such as the DES, DESX, DEAL, FEAL, GOST, Khufu and Khafre, LOKI, CAST, and Blowfish [6], [7]. Rather than the use of many rounds, such as 16 in the DES, Luby and Rackoff introduced a 3-

round FN construction used in designing a provably secure PRP from pseudorandom functions (PRF) [8]. Further analysis and several block ciphers are designed based on the Luby-Rackoff construction [9]–[13]. NLMSFC is a Luby-Rackoff block cipher in which we make use of a new keyed PRF consisting of keyed PR subfunctions in a matrix like structure; the size of this matrix is a data dependent which gives NLMSFC a data dependent structure which significantly strengthens its security. Extensive confusion, diffusion and pseudorandomness tests based on the NIST statistical tests of NLMSFC and its underlying PRF consistently demonstrated their effectiveness. Furthermore, NLMSFC is not practically vulnerable to known attacks. Also it is suitable for both hardware and software implementations.

Although NLMSFC is introduced to be used in the proposed LPRKES, it can be used to strengthen wireless mesh networks clients security by applying it as a candidate with a good pseudorandom and security properties in the well known WPA2 protocol used in IEEE 802.11i standard [14], [15]. In addition, we can exploit the whole scheme (NLMSFC and the LPRKES) to build a smart card based wireless mesh network to enhance its authentication and security in general [16].

The rest of the paper is organized as follows. Section 2 describes the Luby-Rackoff construction in more details, section 3 introduces NLMSFC and its experimental work, section 4 gives the suggested LPRKES with its cryptanalysis, section 5 shows how we can apply NLMSFC in the LPRKES, and section 6 gives the conclusions and future work.

2. PRELIMINARIES

Let "$\oplus$"denote the bit-wise XOR operation and $f_1, f_3 : \{0,1\}^r \rightarrow \{0,1\}^l$ and $f_2 : \{0,1\}^l \rightarrow \{0,1\}^r$ be a keyed PRFs. Given a k-bit key $K \in \{0,1\}^k$, a plaintext message $P = (L,R) \in \{0,1\}^{l+r}$ is divided into an l-bit (left) block L and r-bit (right) block R. Let $C = (U,T) \in \{0,1\}^{l+r}$ be its corresponding ciphertext. In case of l=r (balanced structure), Luby and Rackoff described how to construct a secure (against known / chosen plaintext attacks) PRP $\psi(f_1, f_2, f_3)(L,R) = (U,T)$ over $\{0,1\}^{l+r}$, from r-bit PRF's using a 3-round balanced Feistel network, rather than the use of 16 rounds as in the DES algorithm[8], with U and T computed as follows Fig.1: $S = L \oplus f_1(K_1,R), T = R \oplus f_2(K_2,S)$ and $U = S \oplus f_3(K_3,T)$ where $S,U \in \{0,1\}^l$ and $T \in \{0,1\}^r$. Likewise, $\psi(f_3, f_2, f_1)$ yields the inverse PRP.

Note that because the entropy of the required permutation is (l+r)-bit, at least two rounds of PRFs are needed. But, using two rounds only, the attacker can distinguish the outputs from truly random permutation, if he simply chooses two different inputs with the same R. Luby and Rackoff even suggested the use of 4 rounds to prevent adaptive hosen plaintext-ciphertext attacks. Also unbalanced Luby-Rackoff construction $l \neq r$ is presented [9].

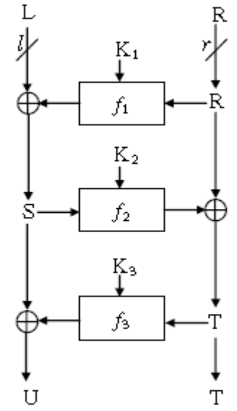


FIGURE 1: Luby-Rackoff cipher construction

3. The Proposed NLMSFC Cipher

As we mentioned, NLMSFC is a balanced 3-round FN cipher (*l=r*) like Luby-Rackoff construction. In addition, the (same) nonlinear matrix structure-based pseudorandom function *f* is employed in each of the three rounds as shown in Fig. 1. The motivation of using this matrix structure cipher is its proven highly diffusion, confusion and security properties [11], [17]. The input to the cipher algorithm is an arbitrary length plaintext that is multiple of 64-bit and an arbitrary length user key UserKey. If the input plaintext length isn't multiple of 64-bit padding will take place to get it multiple of 64-bit before the encryption process.

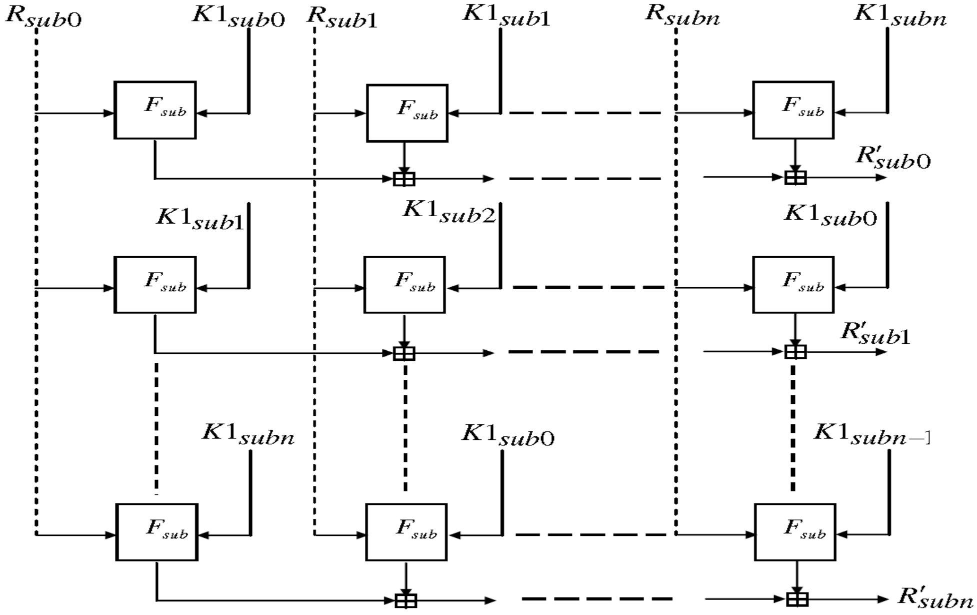


FIGURE 2: The Proposed NLMSFC PR round function F

3.1 The Proposed NLMSFC PR Round Function (F)

The PR round function F uses the PR F_{sub} as its basic building block. The inputs to the round function F are a data block R of length r bits and r-bit round key. First, R is equally divided into n-word [R_{sub0}, R_{sub1}…R_{subn}], each of length 32-bit, also the round key say K_1 is also equally divided into n-word [K_{1sub0}, K_{1sub1}…K_{1subn}]. Then the subfunction F_{sub} will be applied on each these words in a matrix like structure of a data dependent size $n \times n$, as shown in Fig.2, where $\boxplus$ denotes addition mod 2^{32}.

3.2 The Proposed NLMSFC PR Subfunction (Fsub)

F_{sub} is the basic building block in constructing the PR round function F. F_{sub} is an iterated block cipher in which, we perform an invertible keyed operations number of times (rounds) on the input data. F_{sub} is responsible for making the confusion and the diffusion processes required from the PR round function F. Inputs to F_{sub} are a data subblock R_{subi}, $(0 \le i \le n)$ each of length 32-bit and a key subblock K_{subi}, $(0 \le i \le n)$ also each of length 32-bit. F_{sub} performs simple keyed byte operations (addition mod 256, XOR and circular shift left) on the inputs, and it outputs a 32-bit data block.

In designing F_{sub} we take into account that this subfunction must satisfy diffusion and confusion in a minimal number of rounds (4-round).The significant highly nonlinear operation used in F_{sub} is the keyed dependent circular shift used in the 4-round.
Figure 3 shows the PR F_{sub} construction.

The following notations are used in Fig.3.

⊞ Addition mod 2^8
⊕ Bitwise XOR
↲ Circular shift left

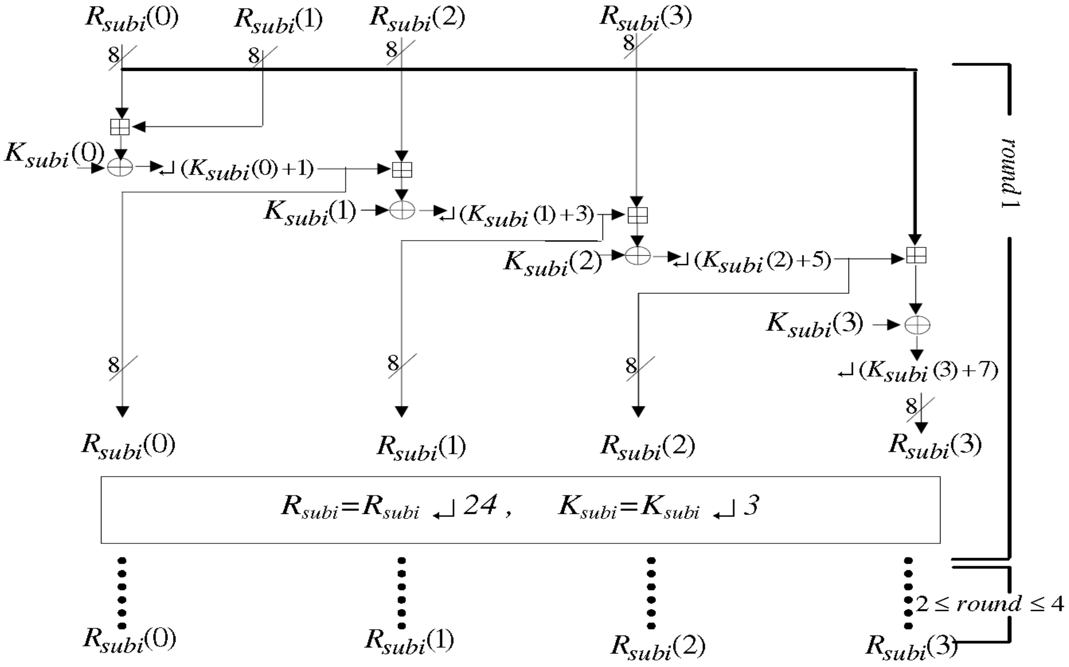


FIGURE 3: The Proposed NLMSFC PR F_{sub}

3.3 The Proposed NLMSFC Key Scheduling Algorithm

The key-scheduling algorithm is used to generate the 3-round keys K_1, K_2 and K_3 each of length r-bit, where the input to the key generation algorithm is the user input key (UserKey), and the output is the UserKey after modifications UserKey =[K_1,K_2,K_3] with length 3r-bit, where 3 indicates that the modified user key UserKey will be equally divided into 3-round keys.
There are 3 cases the key scheduling algorithm handles:
Case $UserKeyLen \geq 3.r$ -bit. In this case the algorithm truncates UserKey to length UserKeyLen = 3r-bit, and then equally divides it into three keys K_1, K_2 and K_3 each of length *r*-bit.
Case $UserKeyLen \geq 64\& < 3.r$ -bit. In this case the algorithm makes expansion to the input UserKey until UserKeyLen=3r-bit, then equally divides it into three keys K_1, K_2 and K_3 each of length *r*-bit.
Case $UserKeyLen < 64$ -bit. In this case padding with ($64 - UserKeyLen$) zeros will take place to the right of UserKey, and then the algorithm makes expansion to UserKey until UserKeyLen=3r-bit, and then equally divides it into the 3-round keys K_1, K_2 and K_3.

Expansion process: The following pseudo code shows the expansion process used in the key scheduling algorithm. In this pseudo code we use the following notations:

$\boxplus$ Addition mod 2^{32}
$\oplus$ Bitwise XOR
$\hookleftarrow$ Circular shift left
| Concatenation

```
1-Index=1
2-Indexx=1
3- While UserKeyLen=3r
        UserKey=UserKey $\hookleftarrow n$
        If Indexx = odd then
            T=UserKey(Index:index+31) $\boxplus$ UserKey(index+32:index+63)
      else
            T=UserKey(Index: index+31) $\oplus$ UserKey(index+32: index+63)
      End if
      UserKey=UserKey | T
      Indexx=Indexx+1
      Index=Index+32
   End while
4-truncate UserKey to length UserKeyLen=3r–bit
```

3.4 NLMSFC Cryptanalysis

In this section, we consider the performance of NLMSFC under several attacks types.

1- Exhaustive key search attack (brut search attack):
In this attack, the attacker has many plaintext-ciphertext pairs encrypted under the same key and his job is to search all possible keys to find the key used in the encryption process. But, NLMSFC prevents such type of attacks through using arbitrary key length, so the attacker cannot practically make such search. In addition, and if we assume that the attacker knows the operating NLMSFC block length B, $B \in \{64,128,192,.....\}$ he must search in $2^{\frac{3B}{2}}$ possible keys (without using the key-scheduling algorithm). So we recommend using large operating block lengths to get such attack computationally infeasible.

2- Dictionary attack: In this attack, the attacker makes a look up table (LUT) containing all possible plaintexts/ciphertexts pairs encrypted under all possible keys. In the case of NLMSFC, we allow the user to encrypt the whole message at once or divide it into blocks of sizes that are multiple of 64 bits (64,128,196,256…). Moreover, we allow the user to use any key length. Then, the attacker neither knows the block size nor the key length used. So he finds no way to make such a dictionary.

3- Linear and Differential Cryptanalysis: In linear and differential attacks [6], the attacker wants to know multiple distinct plaintexts-ciphertexts pairs encrypted under the same key, to know some of the key bits. So in order to prevent these attacks, we can encrypt the whole message at once using a different key each time or simply keep the employed NLMSFC running and use its successive output to encrypt the successive input blocks. However, more analysis needs to be done in this field.

4- Adaptive chosen plaintext/ciphertext attack: The 3-round Luby-Rackoff ciphers may not prevent the adaptive chosen plaintext/ciphertext (two-sided) attack, which is the strongest attack against any symmetric key block cipher (despite being of little practical availability where the attacker can reach both the encryption and decryption engines). So, as suggested by Luby and Rackoff [8], a 4-round NLMSFC successfully prevents such type of attack.

Plaintext	Key	Ciphertext
0000000000000000	0000000000000001	5746958952AD3C9C
0000000000000001	0000000000000000	4B9E731C8A395EB2
0000000000000000	FFFFFFFFFFFFFFFF	AE4E811BB7B07217
FFFFFFFFFFFFFFFF	FFFFFFFFFFFFFFFF	351A6572A06FF9C6
0000000000000001	FFFFFFFFFFFFFFFF	1FB6F2FF51D31232
FFFFFFFFFFFFFFFE	FFFFFFFFFFFFFFFF	BE84D2178229B3FA
FFFFFFFFFFFFFFFF	FFFFFFFFFFFFFFFE	9DA076943DAF1157
FFFFFFFFFFFFFFFF	0000000000000000	E092484DCCB58153

TABLE 1: Examples of 64-bit test vectors (in Hex) for NLMSFC

3.5 NLMSFC Experimental Work

We fully software implemented NLMSFC as a variable block-size variable key-length cipher with a simple effective key scheduling scheme. Table.1 presents examples of plaintext-key-ciphertext NLMSFC test vectors, especially including low and high density and correlated plaintext and key patterns, assuming 64-bit plaintext/key that shows NLMSFC excellent diffusion and confusion properties.

As in all Luby-Rackoff ciphers, security and pseudorandomness of the cipher is based upon the PR of the employed keyed round PRF f_K. The diffusion and confusion properties as well as pseudorandomness of the proposed PRF and the overall NLMSFC have been verified using extensive statistical diffusion and confusion as well as NIST tests [18].

Diffusion Test: 100 64-bit (32-bit for testing the round function) PR plaintexts P_i, i=1,2,.. ..,100 and 100 64-bit key K_i, i=1,2,.. ..,100, are generated using the SEAL algorithm. For each P_i, 64 1-perturbed-bit plaintexts $\{P_{i,j}$, j=1,2,.. ..,64}, with the *jth* bit inverted, are generated. Then, the histogram, mean value and variance of the 6400 hamming distances $d_{i,j}=\sum(E_{Ki}(P_i) \oplus E_{Ki}(P_{i,j}))$ are computed, where E_{Ki} (P_i) means the encryption of plaintext P_i using the K_i key.

Confusion Test: For the $P_{i,j}$'s mentioned above, the histogram, mean value and variance of the 6400 plaintext-ciphertext correlation coefficients $\rho_{i,j}$= corr($P_{i,j}$,$E_{Ki}(P_{i,j})$) are computed. Also, for the P_i's and $P_{i,j}$'s the histogram, mean value and variance of the 6400 ciphertext-ciphertext (of correlated plaintexts) correlation coefficients ρ_{ij} = corr($E_{Ki(Pi,)}$,$E_{Ki}(P_{i,j})$) are computed.

The results of the confusion and diffusion tests (summarized in Table.2 and Fig.4, 5 and 6) illustrate competitive performance compared with the DES and IDEA ciphers [6] as the correlations are almost zero and the percentage of the changing bits due to 1-bit perturbations is almost 50%.

NIST Pseudorandomness tests: The NIST Test Suite is a statistical package composed of 16 tests, basically developed to test the randomness of PRNG sequences. To use the NIST tests for testing the pseudorandomness (and implicitly the diffusion and confusion) of a block cipher, 7 data types are generated, following the procedure suggested in [19]. Of each data type, 100 4096-bit binary sequences were analyzed. These data types include: Plaintext-Avalanche, Key-Avalanche, Plaintext-Ciphertext Correlation, Low-Density Plaintext, Low-Density Key, High-Density Plaintext and High-Density Key data types.

The following 13 tests, with 32 p-values, of the 16 NIST tests were applied, namely the frequency (monobit), frequency within a Block (using a 128-bit block length), runs, longest run-of-1's in a block (using a 128-bit block length), binary matrix rank (with a 3×3 size), discrete Fourier transform, overlapping template matching (using a template of 9 1's, with a block length of 512-bit), Maurer's "universal statistical" (with 4-bit per block with 60 blocks for the initialization sequence), linear complexity (with a 20-bit block length), serial (with a 3-bit block length), approximate entropy (with a 2-bit block length), cumulative sums (Cusums), and random excursions variant tests.

Cipher **Alog**	Diffusion block length=64	Confusion tests block length=64	
		plain /cipher texts Corr.	Ciphertexts Corr.
	mean/64, var/64	Mean, var	Mean, var
NLMSFC	0.50, 0.24	-4.16e-5, 9.57e-4	-6.25e-4, 9.46e-4
DES	0.50, 0.24	-1.05e-5, 9.46e-4	-2.93e-4, 9.67e-4
IDEA	0.50, 0.25	-4.43e-4, 9.65e-4	-6.17e-4, 9.78e-4

TABLE 2: Comparison between the NLMSFC, DES, and IDEA.

Significance level of 0.01 indicates that one would expect 1 sequence out of 100 sequences to be rejected. A p-value ≥ 0.01 means that the sequence can be considered as random with a confidence of 99%. For each p-value, either success or failure evaluation was made based on being either above or below the pre-specified significance level of α=0.01 [18]. For each 100 sequences, two quantities were determined: the proportion of binary sequences passing the statistical test and an extra uniformity p-value based on a chi $\chi 2$ test (with 9 degree of freedom) applied to the p-values of the 100 sequences. A sample (of 100 sequences) was considered to be passed a statistical test if its proportion of success exceeded $(1-\alpha)-\sqrt[3]{\frac{\alpha(1-\alpha)}{m}}=.99-\sqrt[3]{\frac{0.99\times 0.01}{100}}\approx 0.94$, i.e., 94%, and the uniformity test P-value exceeds 0.0001 [18]. The obtained results of the 32 p-values of the NIST tests successfully verified the pseudorandomness, diffusion and confusion properties of the proposed PRF and the overall NLMSFC with more than 94% proportion of succeeded sequences. Figure.7-9 illustrate samples of the obtained results, specifically the proportion of succeeded sequences for the 32 NIST tests applied to NLMSFC with Plaintext-Avalanche, Key-Avalanche, and Plaintext-Ciphertext Correlation generated data types.

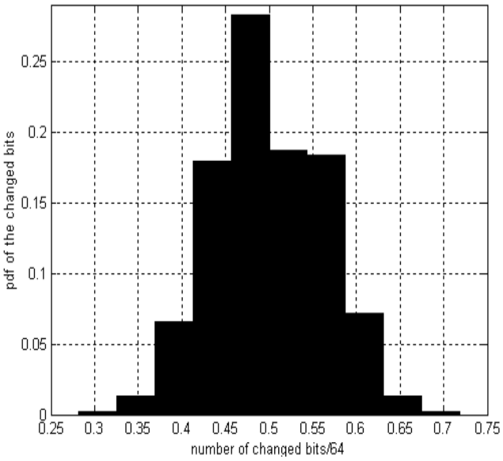


FIGURE 4: Diffusion test: NLMSFC

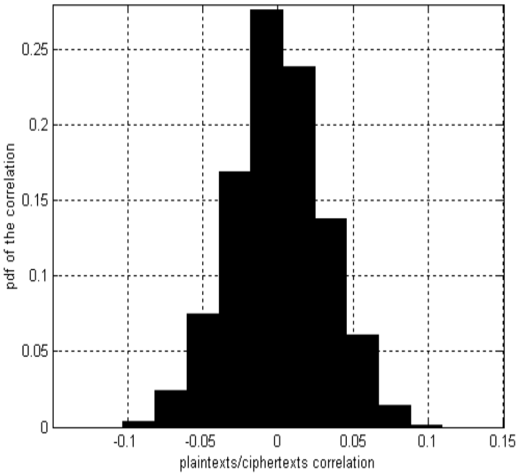


FIGURE 5: Confusion test: NLMSFC plaintext-ciphertexts Correlations histogram

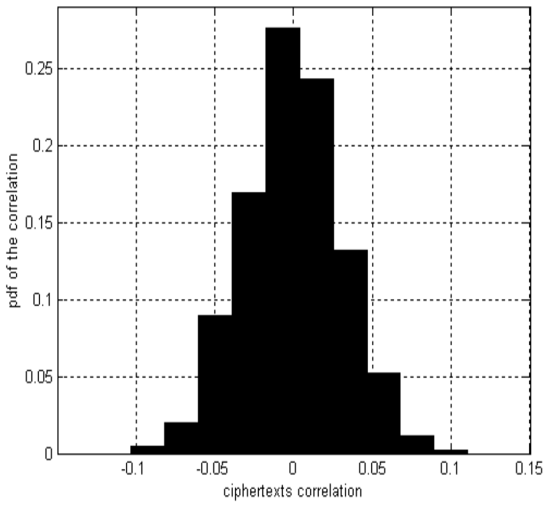


FIGURE 6: Confusion test: NLMSFC ciphertexts Correlations histogram

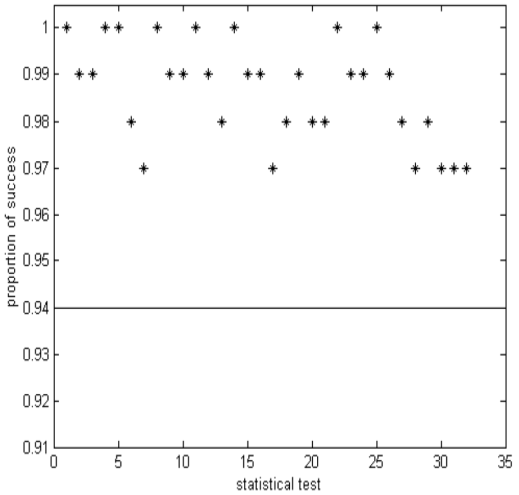


FIGURE 7: NIST tests using Plaintext-Avalanche data: Proportion of succeeded sequences for NLMSFC

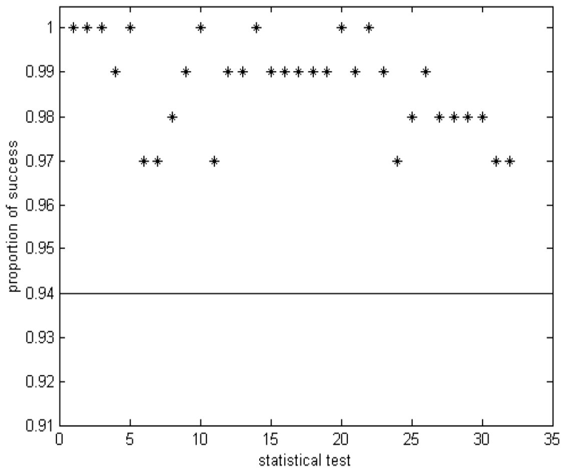


FIGURE 8: NIST tests using Plaintext- Ciphertext correlation: Proportion of succeeded sequences for NLMSFC

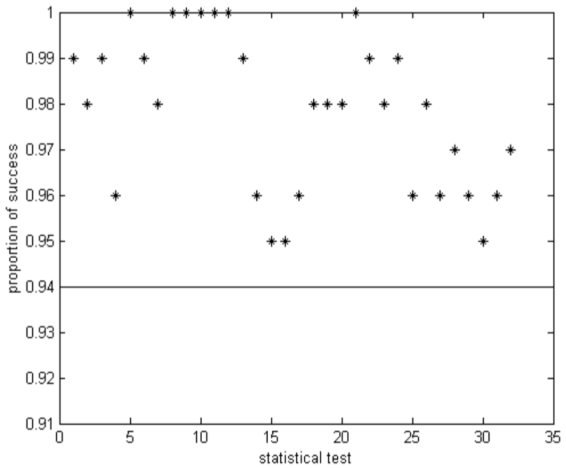


FIGURE 9: NIST tests using key-Avalanche data: Proportion of succeeded sequences for NLMSFC

4. A Novel LPRKES Based upon 2-Round Generalized FN for Smart Cards

4.1 Proposed Generalized 2-Round FN

Since, most of attacks on Luby-Rackoff and multiple rounds FN (e.g., DES) are based upon the linearity properties of the XOR function joining the rounds, we suggest the use of a keyed invertible encryption function $E_K(.):\{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ instead of the XOR to propose a generalized 2-round FN which will be used as a new LPRKES. The E_K/D_K (encryption/decryption) function used in this scheme Fig. 10 should be a strong PRP functions like DES, AES, or the proposed NLMSFC.

In this network, the plaintext P and its ciphertext C is divided into m blocks, i.e. P=(P_1, P_2,….,P_m) and C=(C_1, C_2,….C_m), L=P_1, R=(P_2,…..,P_m), U=C_1, T=(C_2,…,C_m) and *E* denotes the PRP encryption function. $H:\{0,1\}^* \rightarrow \{0,1\}^a$ denotes a collision resistant one way hash function, such as SHA-1 [6], and $F:\{0,1\}^k \times \{0,1\}^a \rightarrow \{0,1\}^k$ is a keyed mapping function (ex , simply XOR). Also, the second round keyed hash function is simply interleaving or concatenating the input with the key, i.e, *H(U|K2), H(K2|U)* or *H(K2|U|K2)*.

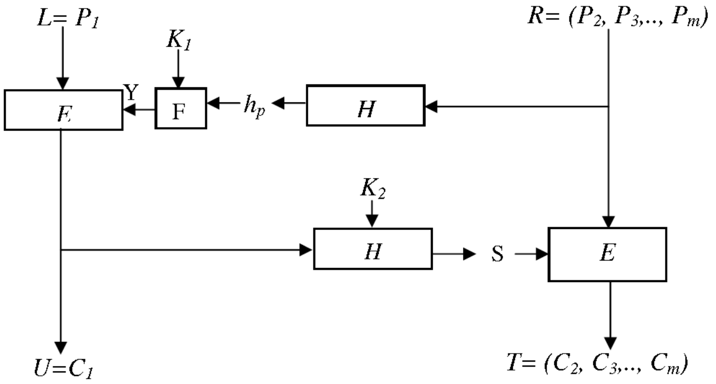


FIGURE 10: The proposed generalized 2-round FN

4.2 A Novel LPRKES Based upon the Proposed Generalized FN

We make use of the above 2-round FN in making a strong and highly secure LPRKES with a one interaction between the host and the smart card which is greatly important from the security point of view.

Proposed LPRKES Encryption Protocol:
Input P= (P$_1$, P$_2$… P$_m$) and output C=(C$_1$, C$_2$….C$_m$).

1. Host: $h_p \leftarrow H(P_2, P_3 ... P_m)$.
2. Host → Card: P_1, h_p.
3. Card: $C_1 \leftarrow E_{F(h_p, K_1)}(P_1)$.
4. Card: $S \leftarrow H(C_1, K_2)$.
5. Card → Host: C$_1$, S.
6. Host: $C_i \leftarrow E_S^i(P_2, P_3 .... P_m)$, $i \in \{2,3,....m\}$.

Proposed LPRKES Decryption protocol:
Input C=(C$_1$, C$_2$….C$_m$) and output P=(P$_1$, P$_2$… P$_m$).

1. Host → Card: C$_1$.
2. Card: $S \leftarrow H(C_1 | K_2)$.
3. Card → Host: S.
4. Host: $P_i \leftarrow D_S^i(C_2, C_3 ... C_m), i \in \{2,3....m\}$.
5. Host: $h_p \leftarrow H(P_2, P_3 ... P_m)$.
6. Host → Card : h_p.
7. Card: $P_1 \leftarrow D_{F(h_p, K_1)}(C_1)$.
8. Card → Host: P$_1$.

4.3 Security Analysis of the proposed LPRKES

We first prove that the proposed LPRKES satisfies LUCKs' postulates [3]:

Theorem 1: the proposed LPRKEs is forgery secure with a probability of $\frac{q^2}{2^l+1}+\varepsilon$, where forgery secure means that if an attacker can execute q encryptions/decryptions with arbitrarily plaintexts/ciphertexts, he can know no more than q valid plaintexts-ciphertexts pairs.

Proof: consider the following two cases with messages M_1= {L_1, R_1} and M_2 = {L_2, R_2}:

Case1: Consider the encryption protocol Fig.10 and assume that $R_1 = R_2$, $L_1 \neq L_2$ and let $E:\{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ be a strong invertible pseudorandom permutation PRP (ex. DES, NLMSFC, AES). Then, the probability Pr (U_1=U_2 for $L_1 \neq L_2$) =0 or almost zero. Consequently, with $U_1 \neq U_2$ and the collision resistance hash function H (i.e., it is infeasible to find $t_1 \neq t_2$ with $H(t_1) = H(t_2)$) [3]. So, Pr ($S_1 = S_2$) will be negligible. Thus, in this case, with a probability near one, the ciphertexts C_1=(U_1,T_1) and C_2=(U_2,T_2) are independently chosen random values. So the attacker has no gain when encrypting many plaintexts with equal right halves. The same analysis will be applied with $L_1 = L_2$, $R_1 \neq R_2$, and for the decryption protocol.

Case 2: Let $L_1 \neq L_2$, $R_1 \neq R_2$. Consequently, $h_{p1} \neq h_{p2}$ and $Y_1 \neq Y_2$ (Fig.10), Also E is a strong PRP which means: Pr (U_1 ($E_{Y1}(L_1)$ = U_2 ($E_{Y2}(L_2)$) ≤ $\frac{1}{2^l}$ +ε where ε is a small number depending on the pseudorandomness of E (if E is truly random then ε = 0). In consequence Pr ($S_1 = S_2$) ≤ $\frac{1}{2^l}$ +ε. If the attacker makes q encryptions, then there are q(q-1)/2 different messages pairs. Thus, the probability of $U_i = U_j$, i ≠ j satisfies {Pr ($U_i = U_j$)} ≤ $\frac{q(q-1)/2}{2^l}$ +ε ≈ $\frac{q^2}{2^{l+1}}$ +ε . The same discussion applies for the decryption protocol. Then, we can conclude that, by observing the encryption/decryption process for q plaintexts/ciphertexts, any attacker can distinguish the encryption/decryption permutation from a truly random permutation with a probability not more than $\frac{q(q-1)/2}{2^l}$ + ε.

Theorem 2: The proposed LPRKES is inversion secure.

Proof: inversion secure means the RKES must prevent chosen plaintext/ciphertext attacks. Such attacks can be done on the proposed scheme with a high probability only if the attacker can simulate the card's part of the RKES. From Theorem 1, the attacker can do this if he is able to encrypt/decrypt about $2^{l/2}$ different plaintexts/ciphertexts using the smart card which is impossible for large *l*. So the proposed LPRKES is inversion secure.

Theorem 3: The proposed LPRKES is pseudorandom.

Proof: The proof is included in proving Theorem 1.

Thus, based on the above analysis and compared to recent RKESs [2-4], the proposed LPRKES has the following advantages:

1. The proposed LPRKES is more secure than Blaze's RKEP [2] because it's shown in [3] that Blaze's RKEP is forgery insecure, inversion insecure, and non-pseudorandom.
2. The proposed LPRKES is more efficient than RaMaRK, because, in RaMaRK, Lucks [3] uses the first two plaintexts blocks in order to define an encryption key for the rest of the message. So any adversary that controls the host during the encryption or decryption of one file of a set of files that start with the same two blocks can subsequently decrypt the encryption of any file in the set. In contrast, the proposed scheme uses the rest of the message to define the key used to encrypt the first plaintext block, and then uses the encryption output of the first block to define the encryption key for the rest of the message. So, the keys used to encrypt two messages will be equal only if the two messages are equal (or after the attacker makes nearly $2^{l/2}$ different encryptions of $2^{l/2}$ different messages).

3. The proposed scheme is more computationally efficient than RaMaRK and BFNLPRKES [4] from the card point of view. In RaMaRK, it is required from the card to evaluate six different PRFs. So it is inadequate for inexpensive smart-cards with limited bandwidth, memory, and processor speed. This also happens in the BFN-LPRKES, in which, it's required from the card to evaluate three encryption functions and three mapping functions. However, the proposed scheme needs from the smart card to evaluate only three functions: encryption, hash and mapping (hash) functions.
4. The proposed scheme is more efficient than the BFN-LPRKES from the host-card communication point of view. The BFN-LPRKES requires two rounds of interaction between the host and the card, but the proposed scheme requires only one round which enhances the security of the scheme.

5. The Application of NLMSFC in the Proposed LPRKES.

Figure.11 shows how we can apply NLMSFC as the PRP in the suggested LPRKES.

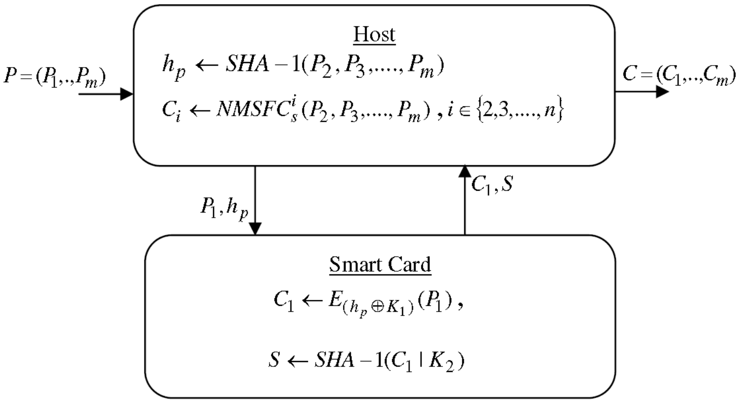

Host
$h_p \leftarrow SHA-1(P_2, P_3, ..., P_m)$
$C_i \leftarrow NMSFC_s^i(P_2, P_3, ..., P_m)$, $i \in \{2,3,....,n\}$
$P=(P_1,..,P_m)$
$C=(C_1,..,C_m)$
P_1, h_p
C_1, S
Smart Card
$C_1 \leftarrow E_{(h_p \oplus K_1)}(P_1)$,
$S \leftarrow SHA-1(C_1 \mid K_2)$

FIGURE 11: The proposed LPRKS using NLMSFC

6. CONSLUSION & FUTURE WORK

This paper deals with cryptographic smart cards protocols which are used to organize the bulk encryption process between the host and the smart card. In an attempt to solve this important issue, we introduce a 2-round network structure, based on a general view of an unbalanced reduced form FN. By exploiting this scheme, we develop smart-card based LPRKES. In addition we analyze this scheme from security and smart card efficiency point of views.
Because the suggested LPRKES is highly depending upon a strong PRP, we also present NLMSFC: A novel Luby-Rackoff construction-based variable block and key lengths symmetric-key block cipher. Its core function is a new pseudorandom function that consists of nonlinear matrix structure with a sub PR function as its elements. Extensive simulations, diffusion, confusion, and NIST pseudorandomess test proof that
NLMSFC and its round function are good PRP and PR function respectively. However, NLMSFC needs a complexity analysis beside the security analysis. But we believe that NLMSFC is less complex.
Also, we show how NLMSFC can be applied as a PRP in the suggested LPRKES. For future development, we will try to apply our cipher and LPRKES in enhancing the security and authentication of the wireless mesh networks especially the wireless backhaul system.

7. REFERENCES

[1] S. Yuan and J. Liu, “Proceedings of the IEEE international conference on e-tech, e-commerce and e-services,” pp.91–110, 2004.

[2] M. Blaze, “High-bandwidth encryption with low-bandwidth smartcards,” Lecture Notes in Computer Science, vol.1039, pp.33–40, 1996.

[3] S. Lucks, “On the security of remotely keyed encryption,” Proceedings of the Fast Software Encryption Workshop, pp.219–229, Springer, 1997.

[4] M. Blaze, J. Feigenbaum, and M. Naor, “A formal treatment of remotely keyed encryption,” Lecture Notes in Computer Science, vol.1403, pp.251–265, 1998.

[5] Y. Hasan, “Key-Joined Block Ciphers with Input-Output Pseudorandom Shuffling Applied to Remotely Keyed Authenticated Encryption,” IEEE International Symposium on Signal Processing and Information Technology, pp.74–79, 2007.

[6] A. Menezes, P. Van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC press, 2001.

[7] A. Biryukov, “Block ciphers and stream ciphers: The state of the art,” Lecture Notes in Computer Science, Proc. COSIC Summer Course, 2003.

[8] M. Luby and C. Rackoff, “How to construct pseudorandom permutations from pseudorandom functions,” SIAM Journal on Computing, vol.17, no.2, pp.373–386, 1988.

[9] M. Naor, “On the Construction of Pseudorandom Permutations: LubyRackoff Revisited,” Journal of Cryptology, vol.12, no.1, pp.29–66, 1999.

[10] R. Anderson and E. Biham, “Two practical and provably secure block ciphers: BEAR and LION,” Lecture Notes in Computer Science, pp.113–120, 1996.

[11] Y. Hasan and E. Mohammed, “PATFC: novel pseudorandom affine transformation-Based Feistel-network cipher,” Signal Processing and Information Technology, 2005. Proceedings of the Fifth IEEE International Symposium on, pp.811–816, 2005.

[12] P. Morin, “A critique of BEAR and LION,” Manuscript, citeseer. nj. nec. Com/124166. html.

[13] Y. Hasan, “YC: A Luby-Rackoff ciphers family driven by pseudorandom vector/matrix transformations,” Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on, pp.1–4, 2007.

[14] S. Frankel, B. Eydt, L. Owens, and K. Kent, “Guide to ieee 802.11 i: Establishing robust security networks,” Technical Report 800-97, National Institute of Standards and Technology Administration US Department of Commerce, Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930, 2006.

[15] F. Martignon, S. Paris, and A. Capone, “MobiSEC: a novel security architecture for wireless mesh networks,” Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks, pp.35–42, ACM New York, NY, USA, 2008.

[16] M. Siddiqui and C. Hong, “Security issues in wireless mesh networks,” IEEE intl. conf. on multimedia and ubiquitous engineering, 2007.

[17] Y. Hasan, “From stream to provably secure block ciphers based on pseudorandom matrix transformations,” Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on, pp.260–265, 2008.

[18] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” , 2001.

[19] J. Soto and L. Bassham, “Randomness Testing of the Advanced Encryption Standard Finalist Candidates. National Institute of Standards and Technology (NIST),” Computer Security Division, 2000.

A Novel Score-Aware Routing Algorithm In Wireless Sensor Networks

Hamid reza Hassaniasl hassaniasl.h@nisoc.ir
Dept. of Compu
Khouzestan Science and Research Branch, IAU
Ahvaz, 61555, Iran

Amir masoud Rahmani rahmani@sr.iau.ac.ir
Dept. of Computer
Tehran Science and Research Branch, IAU
Tehran, 1477893855,Iran

Mashaallah Abbasi Dezfuli abbasi_masha@yahoo.com
Dept. of Computer
Khouzestan Science and Research Branch, IAU
Ahvaz, 61555, Iran

Arash Nasiri Eghbali eghbali@aut.ac.ir
Dept. of Computer engineering
Amir Kabir University
Tehran, Iran

Abstract

Wireless sensor networks are new generation of networks. These networks are appropriate means for collecting and diffusing environmental information, or communicating occurrence of an event to sink node. The main challenge for diffusing information in sensor networks is the limitations in sources and existing bandwidth. Several parameters are involved in selection of middle nodes of information diffusion path, and since existing methods take only some standards into account, it is obvious that they suffer from a number of defects. In our proposed algorithm, five more important factors are combined to score to nodes in routing process. This process called 'SARA[1]' is more flexible, and enhances routing quality through fair division of nodes' roles in transmission of information packets towards sink node. The main objectives of this method are; decrease of consumption energy and the uniform distribution, increase of network lifetime, decrease of traffic load and load balance, possibility of more effective aggregation of data and improving the reliability of formed paths. Several simulations have shown higher competency of this method.

Keywords: Sensor Networks, Routing, Reliability, Load Balance, Lifetime, Score-Aware.

[1] Score-Aware Routing Algorithm

1. INTRODUCTION

Recent years have witnessed considerable advancement in sensor networks. These networks include sensor nodes distributed in the environment. A wireless communication is held between these nodes. Nowadays, there are several applications for sensor networks, and newer applications are still to come. Applications include but not limited to war fields, identifying polluted areas, environmental control, analyzing buildings and roads as well as intelligent highways and medical applications.

The main objective of these networks is collecting and diffusion of environmental information towards sink and it is made through interaction and cooperation between nodes. Therefore, if there is any failure in middle nodes' operation towards sink, the process of information diffusion and ultimately efficiency of network will be reduced. Various factors may cause failure in proper operation of nodes such as termination of nodes' energy. The major aim of all efforts is to overcome difficulties by appropriate approaches which yield to higher efficiency and network lifetime. To do so, a number of information diffusion methods have been created, while each method has had a different approach to each above-mentioned category. Let us review some methods:

The Directed Diffusion (DD) algorithm [1], [2] is a data-centric algorithm for diffusing information in sensor networks. In this algorithm, routing is performed through exchange of local data between neighbor nodes. The mechanism involves diffusion of interest and exploratory packets in opposite direction and reinforcement of selected path. In spite of advantages, this method suffers from defects such as high energy consumption, high traffic, and network division into two separate components.

In [3], two methods are introduced for disjoint and braided multipath routing based on DD algorithm. In braided method, reinforcement packets are sent to some neighboring nodes with higher priority (i.e. nodes that have reached exploratory packet to node sooner), rather than to the first node which has sent exploratory data packet, and thereby multipath are formed towards destination node.

In PCDD algorithm [4], a passive clustering method is proposed to lower consumption energy in DD algorithm. Here, while diffusing exploratory data packets, a tree is formed that in the following execution of algorithm, instead of sending exploratory data packets as a whole, only some part of the tree will be used to send data to the destination node.

The ODCP [5], protocol is proposed to address these two problems: late-aggregation and distinct ED-flooding. In local on-demand clustering protocol, early aggregation and limited ED-flooding can be achieved by using a virtual sink (VS) near the sources. This node plays the role of sink node and broadcasts local interest messages. Therefore the data packets are sent initially to the VS node and then routed toward destination.

In [6], an extension to DD is presented in order to construct multiple paths between the sink and the sensor sources. Using this method, load-balancing is implemented to increase the life-time of the sensor nodes collaborating in the routing process.

The Greedy Multi-path Routing algorithm, [7] is a new localized approach which can produce multiple paths between source and destination nodes. To achieve this, a greedy heuristic method is used in which through implementing efficient load-balancing and energy-aware routing.

In [8], a new mechanism is presented for selecting intermediate nodes for transferring data. It enhances Directed Diffusion algorithm based on nodes' credit.

In this paper, taking a novel approach, we try to introduce an effective method in information diffusion (SARA) so that it optimizes energy consumption and network load distribution with considerable flexibility and assurance. Therefore, parameters such as "closeness to sink", "remaining energy", "sent traffic", "successful sent packets", "number of observable sources" ,that are important to determine middle nodes in information diffusion path, are used. In this

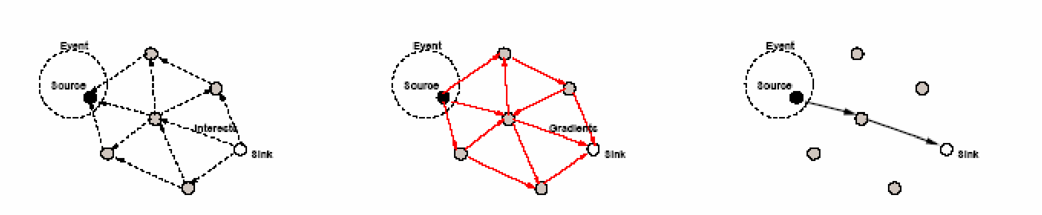
algorithm, during routing phase a series of quotients are used to determine effectiveness of each mentioned parameters and to compute the selection of neighbor node as the next hop.
The paper is structured as follows: Section 2 describes Directed diffusion algorithm and their limitations in wireless sensor networks. The proposed routing algorithm (SARA) is described in section 3. Section 4 outlines our simulation results. Section 5 concludes the paper.

2. OVERVIEW ON DIRECTED DIFFUSION

In this section we review DD [1] as a base method for most routing algorithms.

2.1. Algorithm Procedure

DD algorithm is a data-centric algorithm in sensor networks. To form path, it consists of three stages (Figure 1). In first stage, sink node propagate an interest packet which includes related question. Receiving this packet, each node stores a copy of the packet and sends it to its neighbors. In the second stage, after receiving the packet, the source node floods an exploratory data packet through the network. In this stage, necessary gradients are formed in the network in order to draw the path for information flow. It identifies the direction of information flow and the request statues as well. In the third stage, once exploratory data reach sink node, several paths from source to sink are identified. Sink sends a positive reinforcement packet in opposite direction of a covered path by exploratory data, and thereby reinforces gradient of this single path so that it can be used for sending collected data by sources. This selected path usually has the least delay.



a) Interests propagation b) Initial gradients setup c) Data delivery along reinforced path

Figure 1: A simple schematic for directed Diffusion [1]

2.2. Limitations & Problems

A big problem in this method is that usually the shortest path between source and sink (in terms of the number of hops) is selected. This selection might look reasonable at first glance, but through time the energy of selected path nodes will diminish. As other parameters such as remaining energy of node, processing traffic of node, and node capability in sending information are not considered in selecting path nodes, the problem arises when the path near to the previous path which is the shortest one is selected as the substitute path. Considering the routing method in DD algorithm, it is not far from it. Therefore, if we try to transmit information from one remote point in a quite long time to a sink node, after some time the right and left parts of the network will be divided and they will be disconnected. Hence, this method is not appropriate for data delivery in a continuous manner, making it suitable only for applications with discontinuous diffusion rate.

In DD method, one of high-cost stages of routing is flooding exploratory data by sources throughout the network. This should be conducted for each source, and surplus energy will be imposed on the network. During this operation, due to interference (because of existing common communication medium between nodes), the sent data by sources will be lost. This problem will double with increase of the number of sources and increase of density of nodes in network, and it actually threatens the scalability of DD algorithm.

3. THE PROPOSED ALGORITHM BEHAVIOR (SARA)

To enhance quality of routing in terms of decreasing routing energy, increase of network lifetime, increase of the number of received packets in destination, decrease of delay, and also to enhance reliability of paths formed in DD algorithm, score-aware routing algorithm is introduced. In this algorithm, five different factors are employed, i.e. closeness to sink node, node remaining energy, sent traffic by node in the previous phase, number of successful and fault-free sent packets, and the number of observed sources in one node for scoring to the node during routing process. The nodes with higher score have higher chance for selection as path middle nodes.

In proposed algorithm, based on above-mentioned factors, a number of routing parameters are used to enhance DD routing efficiency. Five quotients w_s ، w_h ، w_e ، w_r ، w_t are used to giving weight to each one of these parameters, and each node in the sensor network will use these quotients during routing to determine the effect of parameters in the calculations for selecting neighbor node as the next hop.

In our developed algorithm, the score given to each node in the network will be calculated as follow:

$$S = F_h \times w_h + F_s \times w_s + F_e \times w_e + F_r \times w_r + F_t \times w_t \quad (1)$$

In this equation, F_s stands for the number of observed sources by a node, F_h corresponds closeness of each node to sink node, F_e is related to remaining energy in each node, F_t stands for sent traffic in each node, and finally F_r indicates assurance degree on relationship between each node and neighboring nodes (link reliability) during one phase of algorithm. While most parameters are transferred through the diffusion of interest packets between the nodes, just the number of observed sources' score is transferred on a limited scale via the exploratory data packets. Based on this method, only the k nodes with higher scores are nominated to receive the exploratory data packets.

3.1. Used Parameters in SARA Algorithm

Considering the algorithm method on routing, each node should consider some score for each neighbor node. This score is determined through calculation of five introduced parameters and their influence on the procedure. The parameters are calculated as below:

3.1.1. F_h Parameter (Distance Between Each Node and Sink)

To send data towards sink node there should be a parameter so that it can be a standard for closeness of node to sink node. To fulfill this requirement, a quotient is used that directly has something to do with the time of reaching interest packet to node. In the following equation, H_n stands for the order of arrival of interest packets to node. The quantity of this parameter is shown by this equation:

$$F_h = 1 - H_n * 0.1 \quad (2)$$

3.1.2. F_s Parameter (Number of Observed Sources by Each Node)

To enhance routing, nodes that are nearer to sources have priority for selection. The number of observed sources by each node will be determined when exploratory data packets are diffused. A distance-from-source label will be put on exploratory data packets, and nodes consider the sum of observed sources with counter-distance quotient as Fs parameter. In the following equation, n_S equal the number of observed sources by node, and d_S equals' number of hops between neighbor node and source node.

$$F_s = \frac{\sum_{i=1}^{n_s}(1/d_{s_i})}{n_s} \quad (3)$$

3.1.3. F_e Parameter (Remaining Energy in Each Node)

One important element in selecting node is node remaining energy, and for routing the priority is with neighbor with higher energy, as the node should have the required energy to send packet. The consequence of this selection is the balance of energy consumption in network nodes. This parameter is calculated based on remaining energy in one node in the start of each phase as below where e_r equals remaining energy, and e_i equals initial energy of node.

$$F_e = {}^{e_r}/_{e_i} \quad (4)$$

3.1.4. F_r Parameter (Reliability of Communication Link)

Parameter F_r shows degree of reliability of neighbor's communication link. The value of this parameter is equal to the proportion of N_s, i.e. sent packets to the total number of received packets (N_r) during one phase of routing algorithm. To simulate the links' reliability at the start of simulation, a P_{drop} between zero and $P_{drop-max}$ as the maximum percent of dropped packets in one communication link is assigned to each node, and accordingly, every packet in every node with possibility of P_{drop} will not be sent. The ultimate value of this parameter is calculated as follows:

$$F_r = \frac{N_s}{N_r} \qquad (5)$$

3.1.5. F_t Parameter (value of Traffic in Each Node)

This parameter shows passing traffic from one node during one phase of algorithm execution. If one node sends more packets during one phase of algorithm execution, in the next phase its chance for selection as the path node will decrease. The following equation shows calculation of F_t parameter.

$$F_t = 1 - \frac{N_s}{N_{max}} \qquad (6)$$

In this equation, N_s stands for the number of sent packets during one phase by one node, N_{max} shows maximum of sent packets during one phase by one source and it is calculated through multiplication of the rate of sent packets by sources during one phase of algorithm execution.

4. SIMULATION RESULTS

To implement algorithm, diffusion 3.20 code along with software NS 2.33 [9] is used.

4.1. Calculation of w quotients

To calculate proportion of w quotients in one scenario, we take the value of w_h as 1, and we consider other quotients zero except one of them, and then calculate the proportion of remaining w to w_h. In all scenarios considered for calculating quotients, a 10*10 grid with average distance of 100 meters between nodes is used.

The simulations' results with varying values assigned to w quotients indicated that the number of received packets in destination reach to their maximum value when w_e equals 0.75, and w_s 3, and w_t 1. Furthermore, to determine the value of w_r we consider the ratio of lost packets to sent ones for each source in a fixed time. Thus the percentage of lost packets for w_r=20 decreased to its lowest value .The following equations are also used to normalize the values of quotients.

$$w_{i-normalaized} = \frac{w_i}{\sum w_i} \qquad i=\{r,s,t,h,e\} \qquad (7)$$

$$\sum w_{i-normalaized} = 1 \qquad (8)$$

4.2. Comparison of SARA Algorithm With DD Algorithm

To compare efficiency of our proposed algorithm with DD, three different scenarios are assumed.

4.2.1. Calculation of Delay Due to Increase of Sources

As shown in Figure 2, the increase of the number of sources leads to a considerable increase of average delay in DD algorithm. However, in SARA algorithm, the average value of delay increases a little. One of the most important reasons for delay decrease in SARA algorithm is deletion of exploratory data packets flooding in each source. Likewise, appropriate load distribution is important in decreasing network nodes' traffic, and thereby in lowering delay for receiving sent packets by source nodes.

As illustrated in the Figure 2, the average delay of received packets in SARA algorithm is about 10 times less than delay in DD algorithm for about 5 sources.

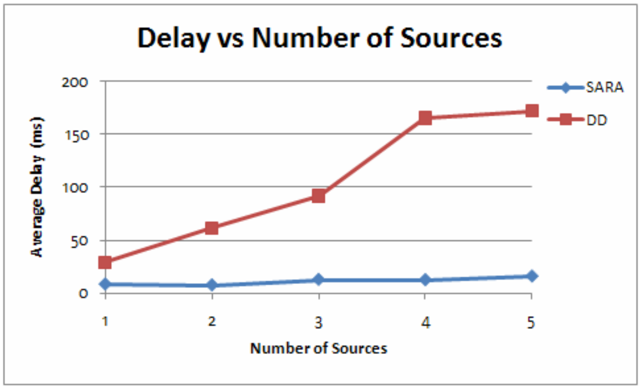


FIGURE 2: Calculation of Delay Due to Increase of Sources

4.2.2. Calculation of Remaining Energy in Nodes Based on Time

As shown in Figure 3, the consumed energy in SARA algorithm is quite half of energy consumption in DD algorithm. The main reasons for decrease of energy consumption are the decrease of exploratory packets and also decrease of traffic in network in SARA algorithm through appropriate load distribution.

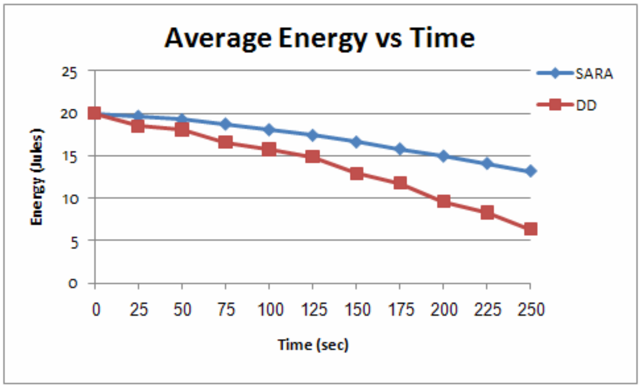


FIGURE 3: Calculation of Remaining Energy of Nodes Based on Time

4.2.3. Calculation of Load Balance in Nodes Due to Increase of Network Density

In Figure 4, to calculate the load balance in nodes, the difference between values of energy of node with the maximum energy with that of the node with the minimum energy is taken as the factor representing load balance. As such, the lower this difference is, the better load balance across network would be.

As illustrated in figure 4, SARA algorithm for higher density and distance between nodes with shorter than 150 meters is more competent than DD algorithm, and as the network density decreases, due to the decrease of possibility of selecting various nodes for path formation, the load balance would decrease.

Likewise, due to the decrease of density, the load balance in DD algorithm firstly decreases, and when the density decrease continues, it goes up until it surpasses SARA algorithm in average distances of grid larger than 150 meters. As mentioned before, the SARA algorithm relies on selection of proper nodes for enhancing load-balancing. However in lower densities, the energy of nodes near the sink is depleted rapidly, leaving no chance for using other energy resources within the network.

Also in lower densities, the broadcasting of exploratory-data in the second phase of algorithm dominates the whole energy usage of the algorithm and affects all of the network nodes. As in SARA algorithm this phase has been omitted and this algorithm has lower energy usage than

DD (about half), the difference between energy of nodes with maximum and minimum energy will be increased in SARA algorithm in this situation.

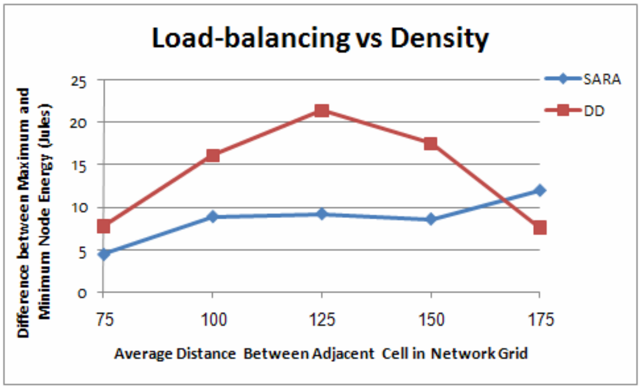


FIGURE 4: Calculation of Load Balance in Nodes Due to Increase of Network Density

5. CONCLUSION & FUTURE WORK

To enhance the routing quality, five different factors are taken into account in the proposed algorithm; the nodes with higher scores have higher chance for selection as the middle nodes of the path.

According to the simulation results, our presented algorithm is more efficient in terms of decreasing delay, decreasing the number of lost packets, improving the load distribution and purposeful network lifetime. It is shown that with higher network density or higher number of sources and higher rate of sent data, the efficiency of our developed algorithm would increase, and such increase is due to the higher number of nodes suitable for selection for routing. Furthermore, since this algorithm unlike DD algorithm, does not need to diffuse exploratory data packets for each source, its scalability will be enhanced considerably.

In future studies, the formal relationship between the parameters and their quotients can be obtained. Then, based on the network status a more precise quotient calculation becomes possible. Forming parallel multi-path routes, by making changes in the algorithm can be considered as a development for the present study.

6. ACKNOWLEDGEMENT

The authors thank NIOC[1] for their sincere supports. We would also like to express our deepest appreciation to NISOC[2].

This study is based on author’s MS thesis entitled: "Improving the Routing in Wireless Sensor Networks via Scoring the Nodes".

7. REFERENCES

[1] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, *“Directed diffusion for wireless sensor networking,”* ACM/IEEE Transactions on Networking, vol. 11, no. 1, pp. 2-16, 2002.

[2] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, *“Building efficient wireless sensor networks with low-level naming,”* In Proceedings of the Symposium on Operating Systems Principles, pp 146-159, October 2001.

[1] National Iranian Oil Company
[2] National Iranian South Oil Company

[3] D. Ganesan, R. Govindan, S. Shenker, D. Estrin. *“Highly resilient energy-efficient multipath routing in wireless sensor networks.”* Proceedings ACM MOBIHOC, pp 251--253, 2001.

[4] V. Handziski, A. K¨opke, H. Karl, C. Frank, W. Drytkiewicz, *“Improving the energy efficiency of directed diffusion using passive clustering,”* European Workshop on Wireless Sensor Networks 2004 (EWSN 2004), pp. 172–187, 2004.

[5] Nasiri Eghbali A., Sanjani H., Dehghan M., *"ODCP: An on-demand clustering protocol for directed diffusion,"* 6th International Conference on AD-HOC Networks & Wireless (AdHocNow 2007), Morelia, Mexico, PP. 218-228, 24-26 Sep., 2007.

[6] Nasiri Eghbali A., Dehghan M., *"Load-balancing using multi-path directed diffusion in wireless sensor networks,"* International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2007), Beijing, China, PP. 44-55, 12-14 Dec., 2007.

[7] Sobhan Masoudi, AmirMasoud Rahmani, Arash Nasiri Eghbali, Ahmad Khademzadeh, *"GMPR: A greedy multi-path routing algorithm for wireless sensor networks,"* fgcn, vol. 1, pp.25-30, 2008 Second International Conference on Future Generation Communication and Networking, 2008

[8] Farnaz Dargahi, AmirMasoud Rahmani, Sam Jabehdari, *"Nodes' credit based directed diffusion for wireless sensor networks,"* International Journal of Grid and Distributed Computing – Vol.1 No.1 December, 2008

[9] NS – network simulator version 2, *http://www.isi.edu/nsnam*

An Efficient Password Security of Multi-Party Key Exchange Protocol based on ECDLP

Jayaprakash Kar jayaprakashkar@yahoo.com
Department of Information Technology
Al Musanna College of Technology
Sultanate of Oman

Banshidhar Majhi bmajhi@nitrkl.ac.in
Department of Computer Science & Engineering
National Institute of Technology
Rourkela, INDIA

Abstract

In this paper we have proposed an efficient password security of multiparty Key Exchange Protocol based on Elliptic Curve Discrete Logarithm Problem. Key exchange protocols allow a group of parties communicating over a public network to establish a common secret key called session key. Due to their significance by in building a secure communication channel, a number of key exchange protocols have been suggested over the years for a variety of settings. Our Protocol is password authentication model, where group member are assumed to hold an individual password rather than a common password. Here we have taken two one-way hash functions to build the level of security high.

Keywords: Key exchange protocol, Password based, secure communication, off-line dictionary attack, ECDLP.

1. Introduction

Group key exchange protocol is an important cryptographic technique in public network, by which a group shares a human-memorable password with a trusted server, can agree a secure session key. Over the past years, many group key exchange protocols have been proposed. However, to our best knowledge, not all of them can meet the requirements of security and efficiency simultaneously. Therefore, in this paper, we would like to propose a new simple multi-party password based authenticated key exchange protocol. Compared with other existing protocols, our proposed protocol does not require any server's public key, but can resist against various known attacks. Therefore, we believe it is suitable for some practical scenarios.

With the proliferation of the hand held wireless information appliances, the ability to perform security functions with limited computing resources has become increasingly important. In mobile devices such as personal digital assistants (PDAs) and multimedia cell phones, the processing resources, memory and power are all very limited, but he need for secure transmission of information may increase due to the vulnerability to attackers of the publicly accessible wireless transmission channel [1]. New smaller and faster security algorithms provide part of the solution, the elliptic curve cryptography ECC provide a faster alternative for public key cryptography. Much smaller key lengths are required with ECC to provide a desired level of security, which means faster key exchange, user authentication, signature generation and verification, in addition to smaller key storage needs. The terms elliptic curve cipher and elliptic curve cryptography refers to an existing generic cryptosystem which use numbers generated from an elliptic curve. Empirical evidence suggests that cryptosystems that utilize number derived from elliptic curve can be more secure [2]. As with all cryptosystems and especially with public-key cryptosystems, it takes years of public evaluation before a reasonable level of confidence in a new system is established. ECC seem to have reached that level now. In the last couple of years, the first commercial implementations are appearing, as toolkits but also in real-world applications, such as email security, web security, smart cards, etc. The security of ECC has not been proven but it is based on the difficulty of computing elliptic curve discrete logarithm in the elliptic curve group [3].

2. Backgrounds

In this section we brief overview of Elliptic Curve over finite field, Elliptic Curve Discrete Logarithm Problem, Key exchange, Elliptic Curve Diffe-Helman (ECDH) and about three-party key exchange protocol.

2.1 The finite field F_P

Let p be a prime number. The finite field F_P is comprised of the set of integers $0,1,2.......p-1$ with the following arithmetic operations [5] [6] [7]:

1. Addition: If $a,b \in F_p$ then $a+b=r$, where r is the remainder when $a+b$ is divided by p and $0 \leq r \leq p-1$. This is known as addition modulo p.
2. Multiplication: If $a,b \in F_p$ then $a.b=s$, where s is the remainder when $a.b$ is divided by p and $0 \leq s \leq p-1.$. This is known as multiplication modulo p.
3. Inversion: If a is a non-zero element in F_P, the inverse of a modulo p, denoted a^{-1}, is the unique integer $c \in F_p$ for which $a.c=1$.

2.2 Elliptic Curve over F_P

Let $p \geq 3$ be a prime number. Let $a,b \in F_p$ be such that $4a^3+27b^2 \neq 0$ in F_P. An elliptic curve E over F_P defined by the parameters a and b is the set of all solutions $(x,y), x,y \in F_p$, to the equation $y^2 = x^3+ax+b$, together with an extra point O, the point at infinity. The set of points $E(F_p)$ forms an abelian group with the following addition rules [9]:

1. Identity: $P+O=O+P=P$, for all $P \in E(F_p)$.
2. Negative : if $P(x,y) \in E(F_p)$ then $(x,y)+(x,-y)=O$, The point $(x,-y)$ is dented as $-P$ called negative of P.

3. Point addition: Let $P(x_1, y_1), Q(x_2, y_2) \in E(F_p)$, then $P + Q = R \in E(F_p)$ and coordinate $(x_{3,} y_3)$ of R is given by $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$ where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$
4. Point doubling : Let $P(x_1, y_1) \in E(F_p)$ where $P \neq -P$ then $2P = (x_3, y_3)$ where $x_3 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1$ and $y_3 = (\frac{3x_1^2 + a}{2y_1})(x_1 - x_3) - y_1$

2.3 Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given an elliptic curve E defined over a finite field F_p ,a point $P \in E(F_p)$ of order n, and a point $Q \in (P)$, find the integer $l \in [0, n-1]$ such that $Q = l.P$. The integer l is called discrete logarithm of Q to base P, denoted $l = \log_p Q$ [9].

2.4 Key exchange

Key exchange protocols allow two parties to agree on a secret shared secret key that they can use to do further encryption for a long message. One of these protocols is the Diffie-Hellman, which is the most used one. The Elliptic curve Diffie- Helman is considered as an extension to the standard Diffie- Hellman.

2.5 Elliptic Curve Diffie-Helman

Elliptic curve Diffie-Helman protocol (ECDH) is one of the key exchange protocols used to establishes a shared key between two parties. ECDH protocol is based on the additive elliptic curve group. ECDH begin by selecting the underlying field F_p or $GF(2^k)$, the curve E with parameters a, b and the base point P . The order of the base point P is equal to n. The standards often suggest that we select an elliptic curve with prime order and therefore any element of the group would be selected and their order will be the prime number n [5]. At the end of the protocol, the communicating parties end up with the same value K which is a point on the curve.

Key exchange

Key exchange protocols allow two parties to agree on a secret shared secret key that they can use to do further encryption for a long message. One of these protocols is the Diffie-Hellman, which is the most used one. The Elliptic curve Diffie-Helman is considered as an extension to the standard Diffie- Hellman. Another direction of research on key agreement is to generalize the two party key agreements to multi party setting.

Group Key Exchange Protocol

Consider the dynamic scenario where participants may join or leave a multicast group at any given time. As a result of the increased popularity of group oriented applications, the design of an efficient authenticated group key agreement protocol has recently received much attention in the literature. A comprehensive treatment have been made to extend the two party (and three party) key agreement protocols to multi party setting. Notable solutions have been suggested by Ingemerson et al. [13], Burmester and Desmedt [10], Steiner et al. [12] and Becker and Willie [11]. All these works assume a passive (eavesdropping) adversary, and the last three provide rigorous proofs of security. For practical applications, efficiency is a critical concern in designing group key agreement in addition to provable security. In particular, number of rounds may be crucial in an environment where numbers of group members are quite large and the group is dynamic. Handling dynamic membership changes get much attention to the current research community. A group key agreement scheme in a dynamic group must ensure that the session key is updated upon every membership change so that subsequent communication sessions are protected from leaving members and previous communication sessions are protected from joining members. Although this can be achieved by running any authenticated group key agreement protocol from scratch whenever group membership changes, alternative approaches to handle this dynamic membership more effectively would be clearly preferable in order to minimize cost of the re-keying operations associated with group updates. The problems of key agreement in Dynamic Peer Groups (DPG) were studied by Steiner et al. [12]. They proposed a class of generic n-party Diffie-Hellman protocols". Atenise et al. [14] [15] introduced authentication into the class of protocols and heuristically analyze their security against active adversary. Steiner et al. [16] consider a number of different scenarios of group membership changes and introduced a complete key management suite CLIQUES studied specially for DPGs which enable addition and exclusion of group members as well as refreshing of the keys. The security analyses of these schemes are heuristic against active adversaries. However, Pereira and Quisquater [20] have described a number of potential attacks, highlighting the need for ways to obtain greater assurance in the security of these protocols. Bresson et al. [17] [18] have recently given a formal security model for group authenticated key agreement. They provided the first provably secure protocols based on the protocols of Steiner et al. [12] for this setting which requires O(n) rounds to establish a key among a group of n users. The initial works [18] [?] respectively consider the static and dynamic case, the security of both of which are in random oracle model following the formalized security model introduced by themselves under the computational Diffie-Hellman (CDH) assumption. They further refine in [18] the existing security model to incorporate major missing details, (e.g. strong corruption and concurrent sessions) and proposed an authenticated dynamic group Diffie-Hellman key agreement proven secure under the DDH assumption within this model. Their security result holds in the standard model instead of random oracle model.

3. Proposed Protocol

Our protocol is designed for use of multi-cast network. The protocol participants consists of a single authenticated server S and multi clients $C_1, C_2 ....C_m$ who wish to establish a session key. All clients have registered their respective password $pw_1, pw_2....pw_m$. Then the multiparty protocol runs among all the clients with the following parameters established:

- Let the elliptic curve E defined over a finite field F_P two field elements $a, b \in F_p$, which defined the equation of the elliptic curve E over F_P i.e. $y^2 = x^3 + ax + b$ in the case $p \geq 3$, where $4a^3 + 27b^2 \neq 0$.
- Let $M_1, M_2 ......M_m$ be m number of group elements in $E(F_p)$.
- Two one-way hash functions G and H , where the output are the elements of F_P
- Iteration Count is the number to be randomly choosed and both the hash function will be executed that numbers of times. Let the number be $c \in [1, n-1]$ $[20]$. So we have to compute both the hash G and H for c no of times.

The proposed protocol follows the follows the following steps.

- **Step -I**: Let each client C_i for $i = 1.2.....m$ selects random numbers $t_i \in [1, n-1]$ and computes the point $P_i = t_i.Q$ and $P_i' = P_i + pw_i.M_i$ and broad cast P_i' to rest of the group.
- **Step -II**: Clients send $(C_1\|P_1^{'})\|(C_2\|P_2')......(C_m\|P'_m)$ to S.
- **Step-III**: Upon receiving, S first recovers P_i by computing $P_i = P_i' - pw_i.M_i$. Next S and R by computing $P = P^{'} - M.pw_A$ and $R = R^{'} - N.pw_B$. Next S select random number u from $[1, n-1]$ and computes $\tilde{P}_i = u.P_i$ for all $i = 1,2.....m$ and then compute the following

$$pw_i'(1) = pw_i.G(C_i\|S\|P_i) \text{ for all } i = 1.2.....m$$
$$pw_i'(2) = G(pw_i'(1))$$
$$\ldots$$
$$pw_i'(c) = G(pw_i'(c-1))$$

Finally we get $pw_i' = G(pw_i'(c))$

Then computes $\tilde{P}'_i = pw_j'.P_i', j = 1,2......m$ and $i \neq j$ and sends $\tilde{P}'_1\|\tilde{P}'_2\|.....\tilde{P}_m^{'}$ to rest of the group.

- **Step -IV** : After having received $\tilde{P}'_1\|\tilde{P}'_2\|.....\tilde{P}_m^{'}$, C_i computes the pair wise key as $K_j = t_j.p\tilde{w}'_j{}^{-1}.(\tilde{P}'_i)$, where $i, j = 1,2.......m$ and $i \neq j$

$\alpha_1 = G(C_1\|C_2\|......\|C_m\|K)$, where $K = K_i = K_j$ for $i, j = 1,2......m$ and $i \neq j$.

$$\alpha_2 = G(\alpha_1)$$
$$\alpha_3 = G(\alpha_2)$$
$$\vdots$$
$$\alpha = \alpha_c.$$

.

Client C_j sends $\tilde{P}'_i\|\alpha$ to C_i for $i, j = 1,2.....m$ and $i \neq j$.

- **Step-V:** With $\tilde{P}'_i\|\alpha$ from C_j, C_i computes $pw'_i = pw_i.G(C_i\|S\|P_i)$, $K_i = t_i.(pw'_i)^{-1}.\tilde{P}'_j$ and verifies that $\alpha = \alpha_c$ by computing $\alpha_1, \alpha_2.......\alpha_c$ and $\alpha_1 = G(C_1\|C_2\|......\|C_m\|K)$ if the verification fails, then C_i aborts the protocol. Otherwise C_i computes the session key SK as

$$SK(1) = H(C_1\|C_2\|........\|C_m\|K)$$

$$\begin{aligned} &SK(2) = H(SK(1)) \\ &\vdots \\ &SK(c) = H(SK(c-1)) \\ &SK = SK(c) \end{aligned}$$

and sends $\beta = \beta_c$, where $\beta_1 = G(C_1 \| C_2 \| .... C_m \| K)$ and $G(\beta_{c-1}) = \beta_c$

- **Step-VI**: Each client C_i verifies the correctness of β is equal to β_c by checking the equation $\beta_1 = G(C_1 \| C_2 \| .... \| C_m \| K)$, $\beta_2 = G(\beta_1) \ldots \beta_c = G(\beta_{c-1})$. If it holds, then each client C_i computes the session key $SK = H(C_1 \| C_2 \| \ldots \| C_m \| \ldots \| K)$, otherwise, C_i abort the protocol.

3.1 Verification of Correctness of 3PAKE

The correctness of the protocol can be verified for each client $C_1, C_2 \cdots C_m$. Let for the client C_1, the key $K_1 = \tilde{P}'_2.(pw_1')^{-1}.t_1$ can be verified with the client C_2 having the key $K_2 = P_1'.(pw_2')^{-1}.t_2$ by computing as

$$K_1 = \tilde{P}_2'.(pw_1')^{-1}.t_1 = (pw_1')^{-1}.(pw_1').\tilde{P}_2.t_1 = u.P_2.t_1 = u.t_1.t_2.Q$$

$$K_2 = \tilde{P}_1'.(pw_2')^{-1}.t_2 = (pw_2')^{-1}.(pw_2').\tilde{P}_1.t_2 = u.P_1.t_2 = u.t_2.t_1.Q$$

Similarly for each client $C_3, C_4 .... C_m$ the correctness of the protocol can be verified.

4. Security discussions

Theorem-1: The protocol does not leak any information that allows verifying the correctness of password guesses.

Proof: Since G is a one-way hash function is executed c times and s, u and t are all random numbers, so the protocol does not leak any information that allow the adversary to verify the correctness of password guesses.

Theorem-2: The protocol is secure against off-line password guessing attacks.

Proof: If the hacker intends to tract out the password, first he has to find out the iteration count c which is a random number and process that number of times. Further he has to solve Elliptic Curve Discrete Logarithm problem (ECDLP) which is computationally infeasible takes fully exponential time. So we can say it is secured against off-line password guessing attacks.

5. Off-Line Dictionary Attack

The proposed protocol is secure against off-line dictionary attacks. This does not leak any information that allows to verify the correctness of password guesses, because G is a one-way function and s, u and t all are random numbers to be taken from $[1, n-1]$. Further the vulnerability of the protocol to the off-line attack can be avoided as

- Consider for the client C_i, let $\overline{pw_i} = G(pw_i)$ and $\overline{pw_j} = G(pw_j)$ for $i \neq j$ and for all $i, j = 1,2 \cdots m$. Then C_i computes $P' = P + \overline{pw_i}.M$ in stead of $P' = P + pw_i.M$, and C_j compute as $R' = R + \overline{pw_j}.N$ instead of as $R' = R + pw_j.N$.
- Accordingly, the Server S recovers P and R is modified to $P = P' - \overline{pw_i}.M$ and $R = R' - \overline{pw_j}.N$.

5. Conclusion

In this research a new protocol for exchanging key between a numbers of parties with a trusted Server has been defined. This new protocol has two major advantages over all previous key exchange protocol, first this protocol does not leak any information that allow the adversary to verify the correctness of password guesses. The second one is that this protocol does not leak any information that allows verifying the correctness of password guesses. The proposed protocol is also easy to implement. The security of our system is based on Elliptic Curve Discrete Logarithm Problem (ECDLP). The primary reason for the attractiveness of ECC over systems such as RSA and DSA is that the best algorithm known for solving the underlying mathematical problem (namely, the ECDLP) takes fully exponential time. In contrast, sub-exponential time algorithms are known for underlying mathematical problems on which RSA and DSA are based, namely the integer factorization (IFP) and the discrete logarithm (DLP) problems. This means that the algorithms for solving the ECDLP become infeasible much more rapidly as the problem size increases than those algorithms for the IFP and DLP. For this reason, ECC offers security equivalent to RSA and DSA while using far smaller key sizes. he attractiveness of ECC will increase relative to other public-key cryptosystems as computing power improvements force a general increase in the key size. The benefits of this higher-strength per-bit include higher speeds, lower power consumption, bandwidth savings, storage efficiencies, and smaller certificates.

References

1. Murat Fiskiran A and B Ruby Lee *"Workload characterization of elliptic curve cryptography and other network security algorithms for constrained environments*". Proc. IEEE Intl.Workshop on Workload Characterization, pp:127-137, 2002.

2. De Win E. and B Preneel "*Elliptic curve public-key cryptosystems - an introduction.State of the Art in Applied Cryptography*", LNCS 1528, pp: 131-141, 1998.

3. Aydos M., E Savas and C .K .KoV 1999. "*Implementing network security protocols based on elliptic curve cryptography*". Proc. fourth Symposium. Computer Networks, pp: 130-139, 1999.

4. Y.F. Chang "*A Practical Three-party Key Exchange Protocol with Round Efficiency*". International Journal of Innovative Computing, Information and Control,Vol.4, No.4, April 2008, 953960.

5. N. Koblitz. "*A course in Number Theory and Cryptography*", 2nd edition Springer-Verlag-1994.

6. K. H Rosen ”*Elementary Number Theory in Science and Communication*”,2nd ed., Springer-Verlag, Berlin, 1986.

7. A. Menezes, P. C Van Oorschot and S. A Vanstone “*Handbook of applied cryptography”*. CRC Press, 1997.

8. D. Hankerson, A .Menezes and S.Vanstone. “*Guide to Elliptic Curve Cryptography* “Springer Verlag, 2004.

9. ”*Certicom ECC Challenge and The Elliptic Curve Cryptosystem*” available: http://www.certicom.com/index.php.

10. M. Burmester and Y. Desmedt ”*A Secure and Efficient Conference Key Distribution System*”. In proceedings of Eurocrypt 1994, LNCS 950, pp. 275-286, Springer-Verlag, 1995.

11. K. Becker and U.Wille ”*Communication Complexity of Group Key Distribution*”. In proceedings of ACM CCS 1998, pp. 1-6, ACM Press, 1998.

12. M. Steiner, G. Tsudik, M. Waidner ”*Diffie-Hellman Key Distribution Extended to Group Communication*”. In proceedings of ACM CCS 1996, pp.31-37, ACM Press, 1996.

13. I. Ingemarsson, D. T. Tang, and C. K. Wong ”*A Conference Key Distribution System*”. In IEEE Transactions on Information Theory 28(5), pp. 714-720, 1982.

14. G. Ateniese, M. Steiner, and G. Tsudik ”*Authenticated Group Key Agreement and Friends*”. In proceedings of ACM CCS 1998[1], pp. 17-26, ACM Press, 1998.

15. G. Ateniese, M. Steiner, and G. Tsudik “*New Multi-party Authenticated Services and Key Agreement Protocols”*. In Journal of Selected Areas in Communications, 18(4), pp. 1-13, IEEE, 2000.

16. M. Steiner, G. Tsudik and M.Waidner ” *Cliques : A New Approach to GroupKey Agreement*” In IEEE Conference on Distributed Computing Systems, May 1998, pp. 380.

17. E. Bresson, O. Chevassut, and D. Pointcheval ” *Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions*”. In proceedings of Eurocrypt 2002, LNCS 2332, pp. 321-336, Springer-Verlag, 2002.

18. E. Bresson, O. Chevassut, and D. Pointcheval. “*Provably Authenticated Group Diffie-Hellman ”Key Exchange - The Dynamic Case*”. In proceedings of Asiacrypt 2001, LNCS 2248, pp. 290-309, Springer-Verlag, 2001.

19. E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater ”*Provably Authenticated Group Diffie-Hellman Key Exchange*”. In proceedings of ACM CCS 2001, pp. 255-264, ACM Press, 2001.

20. Matthew N. Anyanwu, Lih-Yuan Deng and Dipankar Dasgupta "*Design of Cryptographically Strong Generator By Transforming Linearly Generated Sequences"* . In International Journal of Computer Science and Security, pp 186-200, Vol-3, issue-3

21. O. Pereira and J.J. Quisquater"*A Security Analysis of the Cliques Protocol Suite*". In Computer Security Foundations Workshop (CSFW 2001), pp. 73-81, IEEE Computer Society Press, 2001.

22. M. Steiner, G. Tsudik and M.Waidner "Cliques : "*A New Approach to Group Key Agreement*". In IEEE Conference on Distributed Computing Systems, May 1998, pp. 380.

Share Loss Analysis of Internet Traffic Distribution in Computer Networks

D. Shukla diwakarshukla@rediffmail.com
Deptt. of Mathematics and Statistics,
Dr. H.S.Gour Central University, Sagar (M.P.),
470003, INDIA

Virendra Tiwari virugama@gmail.com
Deptt. of Computer Science and Applications
Dr. H.S.Gour Central University, Sagar (M.P.), 470003, INDIA

Sanjay Thakur sanjaymca2002@yahoo.com
Deptt. of Computer Science and Applications
Dr. H.S.Gour Central University, Sagar (M.P.), 470003, INDIA

Arvind Kumar Deshmukh deshmukh.1278@gmail.com
Deptt. of Computer Science and Applications
Dr. H.S.Gour Central University, Sagar (M.P.), 470003, INDIA

Abstract

In present days, the Internet is one of the most required tools of getting information and communicating data. A large number of users through out the world are joining the family of internet in huge proportion. At the same time commercial groups of Internet service provider are also growing in the market. Networks are being overloaded in terms of their capacity and probability of blocking being high day-by-day. This paper presents a share loss analysis of internet traffic when two operators are in competition in respect of quality of service in two markets. The analysis is performed by drawing Iso-share curves through a Markov chain model. The effected over initial traffic share (when final fixed) is examined through simulation study. It is found that network blocking probability highly affects to the initial share amount of traffic of a network operator.

Keywords: Markov chain model, Blocking probability, Call-by-call basis, Internet Service Provider (ISP) [or Operators], Internet traffic, Quality of Service (QoS), Network congestion, Transition probability matrix, Users behavior.

1. INTRODUCTION

Suppose there are two operators (ISP) providing Internet services to people in two markets. Both are in competition to each other in terms of growing more and more to their customer base. Let *p* be initial market share of one operator and *(1-p)* for other. There is another market which has operator O_3 and O_4 with similar initial share of customer base *p* and *(1-p)* respectively. Every operator has tendency to improve upon their customer base constantly. But at the same time they bear constant blocking probability, say L_1 and L_2 in their networks. Because of this fact the quality of services also reduces. This paper presents customer proportion based share loss analysis of Internet Service Providers in two competitive markets when blocking probability increases overtime. The analysis is performed through a probability based Markov Chain model with simulation study of the system.

Markov Chain Model is a technique of exploring the transition behavior of a system. Medhi (1991, 1992) discussed the foundational aspects of Markov chains in the context of stochastic processes. Dorea and Rajas (2004) have shown the application of Markov chain models in data analysis. Shukla and Gadewar(2007) presented a stochastic model for Space Division Switches in Computer Networks. Yuan and lygevers (2005) obtained the stochastic differential equations and proved the criteria of stabilization for Mrakovian switching. Newby and Dagg (2002) presented a maintenance policy for stochastically deteriorating systems, with the average cost criteria. Naldi(2002) performed a Markov chain model based study of internet traffic in the multi-operators environment. Shukla and Thakur (2007, 2008), Shukla, Pathak and Thakur (2007) have shown the use of this kind of model based approach to explain and specify the behavior of internet traffic users. Babikur Mohd. et.al (2009) have shown the flow ased internet traffic classification for bandwidth optimization. Some other useful similar contributions are due to Aggarwal and Kaur (2008), and Agarwal (2009).

2. USER'S BEHAVIOR AND MARKOV CHAIN MODEL

Let O_i and O_j *(i=1,3; j=2,4)* be operators (or ISP) in two competitive locations Market-I and Market–II. Users choose first to a market and then enter into cyber cafe (or shop) situated in that market where computer terminals for operators are available to access the Internet. Let $\{X^{(n)}, n \geq 0\}$ be a Markov chain having transitions over the state space $O_1, O_2, O_3, O_4, R_1, R_2, Z_1, Z_2, A, M_1$ & M_2 where

State O_1: first operator in market-I
State O_2: second operator in market-I
State O_3: third operator in market-II
State O_4: fourth operator in market-II
State R_1: temporary short time rest in market-I
State R_2: temporary short time rest in market-II
State Z_1: success (in connectivity) in market-I
State Z_2: success (in connectivity) in market-II
State A: abandon to call attempt process
State M_1: Market-I
State M_2: Market-II

The $X^{(n)}$ stands for state of random variable X at n^{th} attempt $(n \geq 0)$ made by a user. Some underlying assumptions of the model are:

(a) User first selects the Market-I with probability q and Market-II with probability $(1-q)$ as per ease.
(b) After that User, in a shop, chooses the first operator O_i with probability p or to next O_j with $(1-p)$.
(c) The blocking probability experienced by O_i is L_1 and by O_j is L_2.
(d) Connectivity attempts of User between operators are on call-by-call basis, which means if the call for O_i is blocked in k^{th} attempt $(k>0)$ then in $(k+1)^{th}$ user shifts to O_j. If this also fails, user switches to O_i in $(k+2)^{th}$.
(e) Whenever call connects through either O_i or O_j we say system reaches to the state of success (Z_1, Z_2).
(f) The user can terminate call attempt process, marked as system to abandon state A with probability P_A (either from O_i or from O_j).

(g) If user reaches to rest state R_k ($k=1,2$) from O_i or O_j then in next attempt he may either with a call on O_i or O_j with probability r_k and ($1-r_k$) respectively.

(h) From state R_k user cannot move to states Z_k and A.

The transition diagram is in fig.1 to explain the details of assumptions and symbols. In further discussion, operator $O_1=O_3$ and $O_2=O_4$ is assumed with network blocking parameter $L_1=L_3$, $L_2=L_4$.

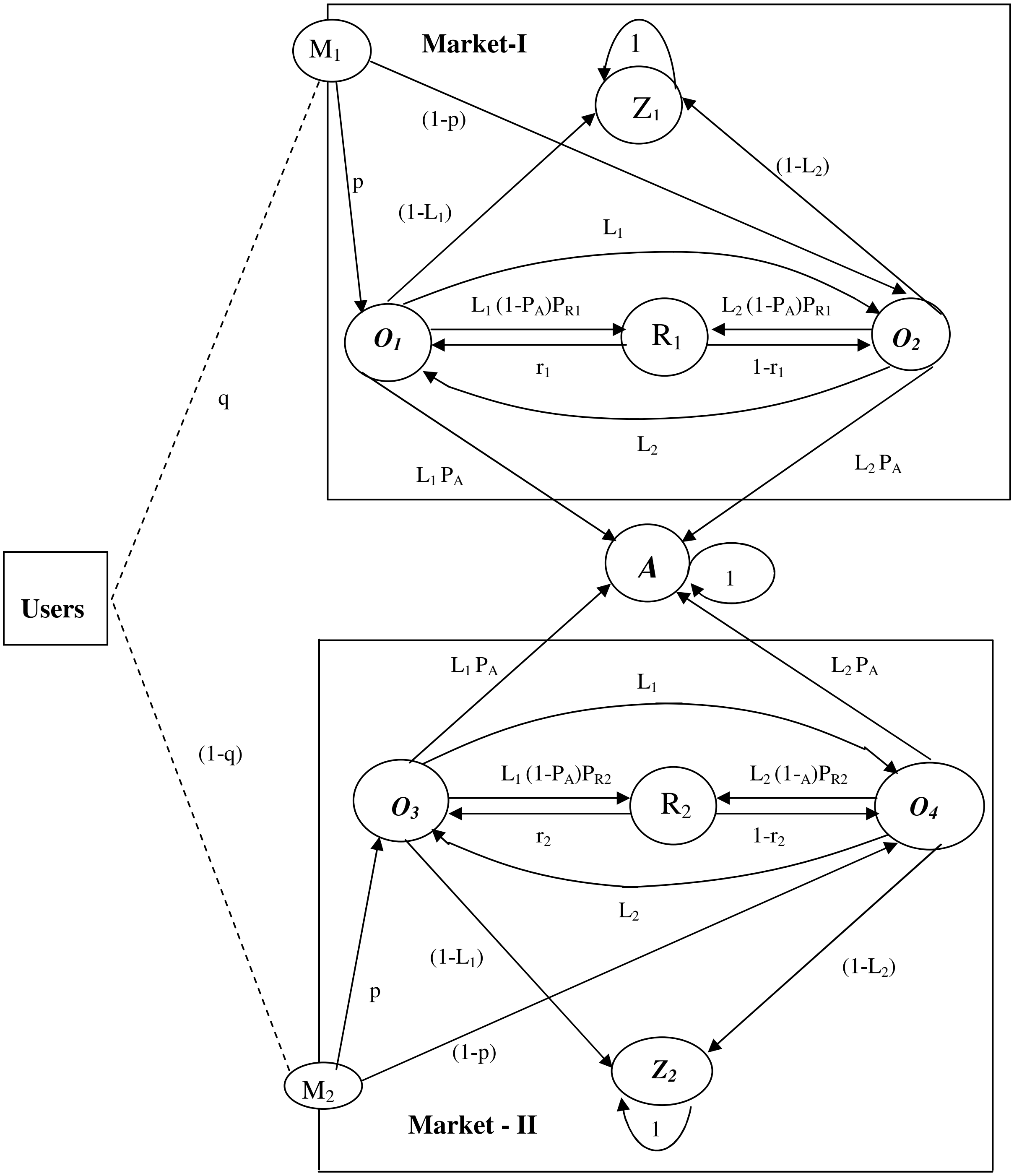


FIGURE 1: Transition Diagram of model.

2.1 The transition probability matrix

States $X^{(n)}$

$X^{(n-1)}$ States

$$
\left[\begin{array}{c|ccccccccccc}
 & O_1 & O_2 & O_3 & O_4 & Z_1 & Z_2 & R_1 & R_2 & A & M_1 & M_2 \\
\hline
O_1 & 0 & \begin{bmatrix} L_1(1-P_A) \\ (1-P_{R_1}) \end{bmatrix} & 0 & 0 & [1-L_1] & 0 & \begin{bmatrix} L_1(1-P_A) \\ P_{R_1} \end{bmatrix} & 0 & [L_1P_A] & 0 & 0 \\
O_2 & \begin{bmatrix} L_2(1-P_A) \\ (1-P_{R_1}) \end{bmatrix} & 0 & 0 & 0 & [1-L_2] & 0 & \begin{bmatrix} L_2(1-P_A) \\ P_{R_1} \end{bmatrix} & 0 & [L_2P_A] & 0 & 0 \\
O_3 & 0 & 0 & 0 & \begin{bmatrix} L_1(1-P_A) \\ (1-P_{R_2}) \end{bmatrix} & 0 & [1-L_1] & 0 & \begin{bmatrix} L_1(1-P_A) \\ P_{R_2} \end{bmatrix} & [L_1P_A] & 0 & 0 \\
O_4 & 0 & 0 & \begin{bmatrix} L_2(1-P_A) \\ (1-P_{R_2}) \end{bmatrix} & 0 & 0 & [1-L_2] & 0 & \begin{bmatrix} L_2(1-P_A) \\ P_{R_2} \end{bmatrix} & [L_2P_A] & 0 & 0 \\
Z_1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
Z_2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
R_1 & r_1 & [1-r_1] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_2 & 0 & 0 & r_2 & [1-r_2] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
M_1 & p & [1-p] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
M_2 & 0 & 0 & p & [1-p] & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
$$

FIGURE 2: Transition Probability Matrix.

2.2 Logic For Transition Probability In Model

(a) The starting conditions (state distribution before the first call attempt) are

$$
\left.\begin{array}{lcl}
P[X^{(0)}=O_1]=0, & and & P[X^{(0)}=O_2]=0, \\
P[X^{(0)}=R_1]=0, & and & P[X^{(0)}=R_2]=0, \\
P[X^{(0)}=Z]=0, & and & P[X^{(0)}=A]=0, \\
P[X^{(0)}=M_1]=q, & and & P[X^{(0)}=M_2]=1-q,
\end{array}\right\} \qquad \ldots(2.2.1)
$$

(b) If in *(n-1)*th attempt, call for O_i is blocked, the user may abandon the process in the n^{th} attempts.

$$P[X^{(n)} = A / X^{(n-1)}=O_i] = P\ [\text{blocked at } O_i].P[\text{abandon the process}]=L_i.P_A \qquad \ldots(2.2.2)$$

Similar for O_j,

$$P[X^{(n)} = A / X^{(n-1)} =O_j] = P\ [\text{blocked at } O_j].P[\text{abandon the process}]=L_j.P_A \qquad \ldots(2.2.3)$$

(c) At O_i in n^{th} attempts call may be made successfully and system reaches to state Z_k from O_i. This happens only when call does not block in $(n-1)^{th}$ attempt

$P[X^{(n)} = Z_k / X^{(n-1)} = O_i] = P[\text{does not blocked at } O_i] = (1-L_i)$ …(2.2.4)

Similar for O_j,

$P[X^{(n)} = Z_k / X^{(n-1)} = O_j] = P[\text{does not blocked at } O_j] = (1-L_j)$ …(2.2.5)

(d) If user is blocked at O_i in (n-1)th attempts, does not want to abandon, then in n^{th} he shifts to operator O_j.

$P[X^{(n)} = O_j / X^{(n-1)} = O_i] = P[\text{ blocked at } O_i].P[\text{does not abandon}] = L_i.(1-p_A)$ …(2.2.5)

Similar for O_j,

$P[X^{(n)} = O_i / X^{(n-1)} = O_j] = P[\text{ blocked at } O_j].P[\text{does not abandon}] = L_j.(1-p_A)$ …(2.2.6)

(e) For operator O_i.

$P[X^{(n)} = O_i / X^{(n-1)} = R_k] = r_k.$ …(2.2.7)

Similar for O_j,

$P[X^{(n)} = O_j / X^{(n-1)} = R_k] = 1-r_k.$ …(2.2.8)

(f) For M_k, (k=1,2) for O_i, O_j

$P[X^{(n)} = O_i / X^{(n-1)} = M_k] = p.$ …(2.2.9)

Similar for O_j,

$P[X^{(n)} = O_j / X^{(n-1)} = M_k] = 1-p.$ …(2.2.10)

3. CATEGORIES OF USERS

Define three types of users as

(i) Faithful User (FU).
(ii) Partially Impatient User (PIU).
(iii) Completely Impatient User (CIU).

4. SOME RESULTS FOR n^{th} ATTEMPTS

At n^{th} attempt, the probability of resulting state is derived in following theorems for all $n=0,1,2,3,4,5....$ for market-I.

THEOREM 4.1: If user is FU and restrict to only O_1 and R_1 in M_1 then n^{th} step transitions probability is

$$\left.\begin{aligned} P[X^{(2n)} = O_1] &= \left[pL_1{}^n(1-p_A)^n p_{R1}{}^n r_1{}^n\right] \\ P[X^{(2n+1)} = O_1] &= \left[qpL_1{}^n(1-p_A)^n p_{R1}{}^n r^n\right] \end{aligned}\right\} \qquad ..(4.1.1)$$

THEOREM 4.2: If user is FU and restrict to only O_2 and R_1 then n^{th} step transitions probability is

$$\left.\begin{aligned} P[X^{(2n)} = O_2] &= \left[(1-p)L_2{}^n(1-p_A)^n p_{R1}{}^n(1-r_1)^n\right] \\ P[X^{(2n+1)} = O_2] &= \left[q(1-p)L_2{}^n(1-p_A)^n p_{R1}{}^n(1-r_1)^n\right] \end{aligned}\right\} \qquad …(4.1.2)$$

THEOREM 4.3: If user is PIU and restricts to attempt between O_1 and O_2 and not interested to state R in M_1 then

$$\begin{aligned}
P[X^{(2n)} = O_1] &= \left[q(1-p)L_1^{(n-1)}L_2^{(n)}(1-p_A)^{(2n-1)}(1-p_{R_1})^{(2n-1)}\right]\\
P[X^{(2n+1)} = O_1] &= \left[qpL_1^{(n)}L_2^{(n)}(1-p_A)^{(2n)}(1-p_{R_1})^{2(n)}\right]\\
P[X^{(2n)} = O_2] &= \left[qpL_1^{(n)}L_2^{(n-1)}(1-p_A)^{(2n-1)}(1-p_{R_1})^{(2n-1)}\right]\\
P[X^{(2n+1)} = O_2] &= \left[q(1-p)L_1^{(n)}L_2^{(n)}(1-p_A)^{(2n)}(1-p_{R_1})^{(2n)}\right]
\end{aligned} \qquad \ldots(4.1.3)$$

THEOREM 4.4: If user is CIU and attempts among *O*$_1$, *O*$_2$ and *R* only in *M*$_1$ then at n^{th} attempt the approximate probability expression are

$$\begin{aligned}
P[X^{(2n)} = O_1] &= \left[q(1-p)L_1^{(n-1)}L_2^{(n)}(1-p_A)^{(2n-1)}(1-p_{R_1})^{(2n-1)}\right]\\
&\quad +\left[pL_1^{(n)}L_2^{(n-1)}(1-p_A)^{(2n-1)}(1-p_{R_1})^{(2n-2)}\,p_{R_1}r_1\right]\\
P[X^{(2n+1)} = O_1] &= \left[qp\,.L_1^{n}L_2^{n}(1-p_A)^{2n}(1-p_{R_1})^{2n}\right]\\
&\quad +\left[(1-p).L_1^{(n-1)}L_2^{(n+1)}(1-p_A)^{2n}(1-p_{R_1})^{(2n-1)}\,p_{R_1}.(1-r_1)\right]\\
P[X^{(2n)} = O_2] &= \left[qp\,.L_1^{(n)}L_2^{(n-1)}(1-p_A)^{(2n-1)}(1-p_{R_1})^{(2n-1)}\right]\\
&\quad +\left[(1-p).L_1^{(n-1)}L_2^{(n)}(1-p_A)^{(2n-1)}(1-p_{R_1})^{(2n-2)}\,p_{R_1}.(1-r_1)\right]\\
P[X^{(2n+1)} = O_2] &= \left[q(1-p)L_1^{n}L_2^{n}(1-p_A)^{2n}(1-p_{R_1})^{2n}\right]\\
&\quad +\left[pL_1^{(n+1)}L_2^{(n-1)}(1-p_A)^{2n}(1-p_{R_1})^{(2n-1)}\,p_{R_1}r_1\right]
\end{aligned} \qquad \ldots(4.1.4)$$

5. TRAFFIC SHARING AND CALL CONNECTION

The traffic is shared between *O*$_i$ *and O*$_j$ operators. Aim is to calculate the probability of completion of a call with the assumption that it is achieved at n^{th} attempt with operator *O*$_i$ (i =1, 3) in market *M*$_1$.

$\overline{P_1}^{(n)}$ =*P[call completes in n*th *attempt with operator O*$_1$*] = P[at (n-1)*th *attempt user is on O*$_1$*]. P[user is at Z in n*th *attempt when was at O*$_1$ *in (n-1)*th*]*

$$\overline{P_1}^{(n)} = P\left[X^{(n-1)} = O_1\right]P\left[X^{(n)} = Z/X^{(n-1)} = O_1\right] = (1\text{-}\mathrm{L}_1)\left[\sum_{\substack{i=0\\ i=even}}^{n-1} P\left[X^{(i)} = O_1\right] + \sum_{\substack{i=0\\ i=odd}}^{n-1} P\left[X^{(i)} = O_1\right]\right]$$

Similarly for operator *O*$_2$

$$\overline{P_2}^{(n)} = P\left[X^{(n-1)} = O_2\right]P\left[X^{(n)} = Z/X^{(n-1)} = O_2\right] = (1-L_2)\left[\sum_{\substack{i=0\\ i=even}}^{n-1} P\left[X^{(i)} = O_2\right] + \sum_{\substack{i=0\\ i=odd}}^{n-1} P\left[X^{(i)} = O_2\right]\right]$$

This could be extended for all three categories of users.

(A) TRAFFIC SHARE BY FAITHFUL USERS (FU)

The FU are those who are hardcore to an operator and never think about others to take services. Using expression (4.1.1) we write for *M*$_1$

$$\left[\overline{P_1}^{(n)}\right]_{FU} = (1\text{-}\mathrm{L}_1)\left[\sum_{\substack{i=0\\ i=even}}^{n-1} P\left[X^{(i)} = O_1\right] + \sum_{\substack{i=0\\ i=odd}}^{n-1} P\left[X^{(i)} = O_1\right]\right] \text{ Under (4.1.1), (4.1.2)}$$

Let $A = \left[L_1 (1 - P_A) P_{R_1} r_1 \right]$, $B = \left[L_2 (1-P_A) P_{R_1} (1-r_1) \right]$, $C = \left[L_1 L_2 (1-P_A)^2 (1-P_{R_1})^2 \right]$,

$D = \left[L_1{}^2 L_2 (1-P_A)^3 (1-P_{R_1})^2 P_{R_1} \right]$, $E = \left[L_2{}^2 (1-P_A)^2 (1-P_{R_1}) P_{R_1} \right]$

For operator O_1, final traffic share by FU

$$\left[\overline{P_1}^{(2n)} \right]_{FU} = (1-L_1).p \left\{ \frac{1-\left[A^2\right]^{n-1}}{1-\left[A^2\right]} \right\} + (1-L_1).qp[A] \left\{ \frac{1-\left[A^2\right]^{n}}{1-\left[A^2\right]} \right\}$$

$$\left[\overline{P_1}^{(2n+1)} \right]_{FU} = (1-L_1).p \left\{ \frac{1-\left[A^2\right]^{(n)}}{1-\left[A^2\right]} \right\} + (1-L_1).qp[A] \left\{ \frac{1-\left[A^2\right]^{(n-1)}}{1-\left[A^2\right]} \right\}$$

Final traffic share for operator O_2 using (4.1.2)

$$\left[\overline{P_2}^{(2n)} \right]_{FU} = (1-L_2).(1-p) \left\{ \frac{1-\left[B^2\right]^{(n-1)}}{1-\left[B^2\right]} \right\} + (1-L_2).q(1-p)[B] \left\{ \frac{1-\left[B^2\right]^{n}}{1-\left[B^2\right]} \right\}$$

$$\left[\overline{P_2}^{(2n+1)} \right]_{FU} = (1-L_2).(1-p) \left\{ \frac{1-\left[B^2\right]^{(n)}}{1-\left[B^2\right]} \right\} + (1-L_2).q(1-p)[B] \left\{ \frac{1-\left[B^2\right]^{(n-1)}}{1-\left[B^2\right]} \right\}$$

(B) TRAFFIC SHARE BY PARTIALLY IMPATIENT USERS (PIU)

The PIU are those who only toggles between operators O_i and O_j but do not want temporary rest (not to chose R_k state). Using expression (4.1.3) for M_1

$$\left[\overline{P_1}^{(n)} \right]_{PIU} = (1 - \mathrm{L}_1) \left[\sum_{\substack{i=0 \\ i=even}}^{n-1} P\left[X^{(i)} = O_1\right] + \sum_{\substack{i=0 \\ i=odd}}^{n-1} P\left[X^{(i)} = O_1\right] \right]$$ Under theorem 4.1.3.

Final traffic share for operator O_1

$$\left[\overline{P_1}^{(2n)} \right]_{PIU} = (1-L_1).p \left\{ 1 + q[C] \frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]} \right\} + (1-L_1).qp \left\{ [C] \frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]} \right\}$$

$$\left[\overline{P_1}^{(2n+1)} \right]_{PIU} = (1-L_1).p \left\{ 1 + q[C] \left\{ \frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]} \right\} \right\} + (1-L_1).qp \left\{ [C] \left\{ \frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]} \right\} \right\}$$

Final traffic share for operator O_2

$$\left[\overline{P_2}^{(2n)}\right]_{PIU} = (1-L_2)(1-p)\left\{1+q[C]\left\{\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right\}\right\} + (1-L_2).q(1-p)\left\{[C]\left\{\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right\}\right\}$$

$$\left[\overline{P_2}^{(2n+1)}\right]_{PIU} = (1-L_2).(1-p)\left\{1+q[C].\left\{\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right\}\right\} + (1-L_2).q(1-p)\left\{[C]\left\{\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right\}\right\}$$

(C) TRAFFIC SHARE BY COMPLETELY IMPATIENT USERS (CIU).

The CIU are those who transit among O_i, O_j *and* R_k . Then using expression (4.1.4) we write for M_1

$$\left[\overline{P_1}^{(n)}\right]_{CIU} = (1-\mathrm{L}_1)\left[\sum_{\substack{i=0\\ i=even}}^{n-1} P\left[X^{(i)}=O_1\right] + \sum_{\substack{i=0\\ i=odd}}^{n-1} P\left[X^{(i)}=O_1\right]\right] \quad \text{Under theorem 4.1.4}$$

Final traffic share for operator O_1

$$\left[\overline{P_1}^{(2n)}\right]_{CIU} = (1-L_1)p\left\{1+q[C]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right] + \left[D\ r_1\right]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right]\right\}$$

$$+(1-L_1)\left\{qL_2(1-P_A)(1-P_{R_1})[C]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right] + \left[E\,(1-r_1)\right]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right]\right\}$$

$$\left[\overline{P_1}^{(2n+1)}\right]_{CIU} = (1-L_1).p\left\{1+q[C]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right] + \left[D\ r_1\right]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right]\right\}$$

$$+(1-L_1)\left\{qL_2(1-P_A)(1-P_{R_1})[C]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right] + \left[E\,(1-r_1)\right]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right]\right\}$$

Final traffic share for operator O_2

$$\left[\overline{P_2}^{(2n)}\right]_{CIU} = (1-L_2)p\left\{1+q[C]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right] + \left[D\,(1-r_1)\right]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right]\right\}$$

$$+(1-L_2)\left\{qL_2(1-P_A)(1-P_{R_1})[C]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right] + \left[E\ r_1\right]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right]\right\}$$

$$\left[\overline{P_2}^{(2n+1)}\right]_{CIU} = (1-L_2).(1-p)\left\{1+q[C]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right]+\left[D(1-r_1)\right]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right]\right\}$$
$$+(1-L_2)\left\{qL_2(1-P_A)(1-P_{R_1})[C]\left[\frac{1-\left[C^2\right]^{(n-1)}}{1-\left[C^2\right]}\right]+\left[E\,r_1\right]\left[\frac{1-\left[C^2\right]^{(n)}}{1-\left[C^2\right]}\right]\right\}$$

6. BEHAVIOR OVER LARGE NUMBER OF ATTEMPTS

Suppose *n* is very large, then $\overline{P_k} = \left[\lim_{n\to\infty} \overline{P_k}^{(n)}\right]$, *k=1, 2* and we get final traffic shares,

$$\left[\overline{P_1}\right]_{FU} = \left\{\frac{(1-L_1).p}{1-\left[A^2\right]}\right\}+\left\{\frac{(1-L_1).qp\,[A]}{1-\left[A^2\right]}\right\}$$

$$\left[\overline{P_2}\right]_{FU} = \left\{\frac{(1-L_2).(1-p)}{1-\left[B^2\right]}\right\}+\left\{\frac{(1-L_2).q(1-p)[B]}{1-\left[B^2\right]}\right\}$$

$$\left[\overline{P_1}\right]_{PIU} = \left\{(1-L_1).p+\frac{(1-L_1).pq[C]}{1-\left[C^2\right]}\right\}+\left\{\frac{(1-L_1).qp[C]}{1-\left[C^2\right]}\right\}$$

$$\left[\overline{P_2}\right]_{PIU} = (1-L_2)(1-p)+\left\{\frac{(1-L_2)(1-p)q[C]}{1-\left[C^2\right]}\right\}+\left\{\frac{(1-L_2).q(1-p)[C]}{1-\left[C^2\right]}\right\}$$

$$\left[\overline{P_1}\right]_{CIU} = (1-L_1)p\left\{1+\left[\frac{q[C]}{1-\left[C^2\right]}\right]+\left[\frac{\left[D\,r_1\right]}{1-\left[C^2\right]}\right]\right\}+\left\{\left[\frac{q(1-L_1)L_2(1-P_A)(1-P_{R_1})[C]}{1-\left[C^2\right]}\right]+\left[\frac{(1-L_1)(1-r_1)[E\,]}{1-\left[C^2\right]}\right]\right\}$$

$$\left[\overline{P_2}\right]_{CIU} = (1-L_2)p\left\{1+\left[\frac{q[C]}{1-\left[C^2\right]}\right]+\left[\frac{\left[D(1-r_1)\right]}{1-\left[C^2\right]}\right]\right\}+\left\{\left[\frac{q(1-L_2)L_2(1-P_A)(1-P_{R_1})[C]}{1-\left[C^2\right]}\right]+\left[\frac{(1-L_2)r_1[E]}{1-\left[C^2\right]}\right]\right\}$$

7. TRAFFIC SHARE LOSS ANALYSIS

Share loss relates to the imbalance between the initial share and final share of traffic between the two operators. Defining loss Δ*p* as the difference between the initial share of O_1 and the final share, derived by the theorem of Faithful User (FU), Partially Impatient User (PIU) and Completely Impatient User (CIU) for O_1

$$\Delta p = p - \overline{P_1}$$

$$[\Delta p]_{FU} = \frac{\left\{p(1-A^2)-p(1-L_1)+qpL_1(1-L_1)(1-P_A)P_{R_1}r_1\right\}}{1-A^2}$$

$$[\Delta p]_{PIU} = \frac{\left\{ pL_1\left[1 - C^2\right] - \left[2qp(1 - L_1).C\right]\right\}}{\left[1 - C^2\right]}$$

$$[\Delta p]_{CIU} = \frac{\left\{ p(2-L_1)\left[1-C^2\right] + \left[q(1-L_1).C\left[p + L_2(1-P_A)(1-P_{R_1})\right]\right] + \left[p(1-L_1)D.r_1 + E(1-L_1)(1-r_1)\right]\right\}}{\left[1-C^2\right]}$$

If Δp negative, means the operator O_1 has actually benefited from the repeated call attempts and has increased its traffic share beyond to its initial expectation *p.* If Δp is positive then operator O_1 has loss of traffic due to blocking and frequent call attempts.

8. INITIAL SHARE ANALYSIS

A. BY FU :

In fig. 8.1 and 8.2, this is to observe that final traffic share (for fixed initial share) has variation over increasing self blocking in the form of linear pattern. For maintaining 70% initial share of FU's operator O_1 has to keep blocking below 40%.

The P_R probabilities of rest state doesnot affect the loss of traffic of O_1. With the 50% of initial customer base and 25% blocking chances, the final customer share is likely to be nearly 15%.

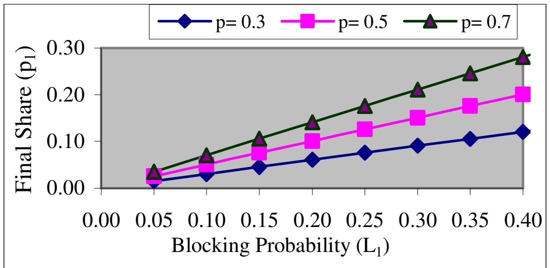


Fig. 8.1 (q=0.4, P_{R1}=0.1, P_A=0.6, r1=0.3 for FU User Iso-Share Curves)

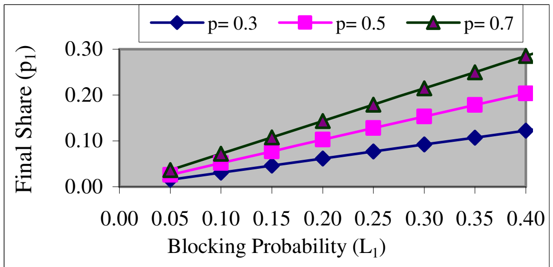


Fig. 8.2 (q=0.4, P_{R1}=0.7, P_A=0.6, r1=0.3 for FU User Iso-Share Curves)

When fig. 8.3-8.4 are underway, it seems that with increasing L_1 along with P_R probability, the final share has line based pattern. But when transition from rest state to O_1 increases (r_1), the proportion of final share by FU improves. So, increasing P_R and *r* simultaneously uplifts the final traffic of the operators. Both probabilities P_R and r have paritive impact over the final share of operator O_1.

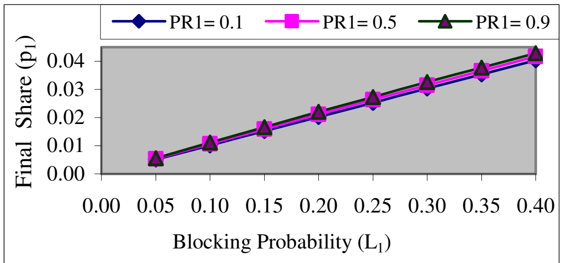


Fig. 8.3 (p=0.1, q=0.9, P_A=0.1, r1=0.2 for FU User Iso-Share Curves)

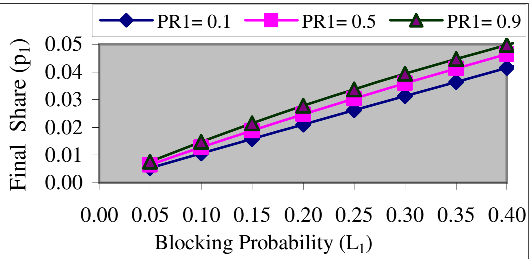


Fig. 8.4 (p=0.1, q=0.9, P_A=0.1, r1=0.5 for FU User Iso-Share Curves)

B. BY PIU :

By fig. 8.5 and fig. 8.6 with the increasing L_1 the final share loss of operator O_1 gets high. But when transition from operator O_1 (P_R) is high the proportion of PIU users is more so the final share loss of operator is higher with the variation of P_R probabilities.

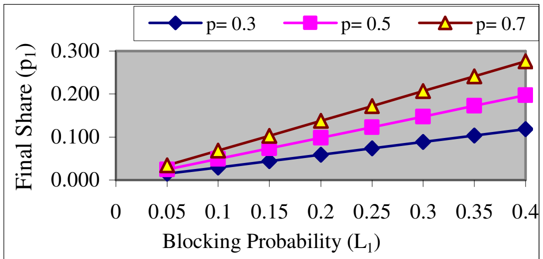


Fig. 8.5 (L_2=0.5, q=0.4, P_{R1}=0.7, P_A=0.2 for PIU User Iso-Share Curves)

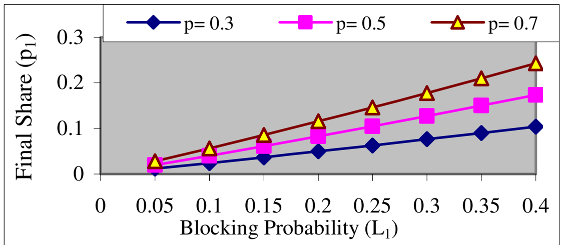


Fig. 8.6 (L_2=0.5, q=0.4, P_{R1}=0.08, P_A=0.2 for PIU User Iso-Share Curves)

For maintaining the 70% initial share operator O_1 has to compensate with 20% final share loss at 30% blocking probability. When P_R probability exceeds for maintaining the same level share operator has 25% initial share loss. This loss has to be compensate by operator because his PIU user proportions is decreased due to more P_R probability.

As per fig 8.7 and 8.8 final share loss with the variation of L_2 over the P_R probability has a downward trend. With increase of only P_R the final traffic share is relatively high. But when self blocking of operator O_1 is high with the opponent blocking then this initial share proposition improves with increasing P_R.

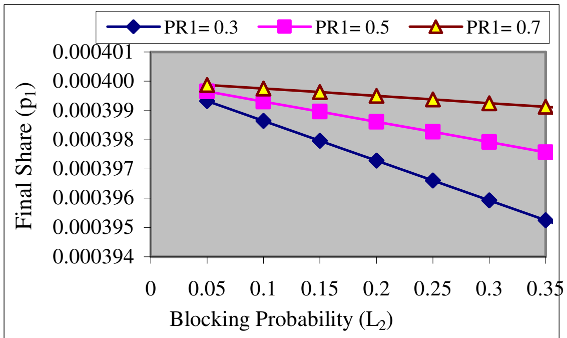


Fig. 8.7 (L_1=0.02, q=0.04, p=0.02, P_A=0.06 for PIU User Iso-Share Curves)

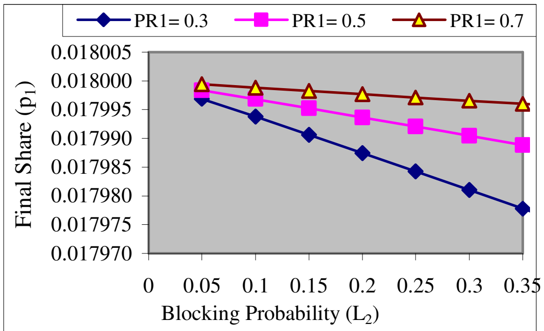


Fig. 8.8 (L_1=0.9, q=0.04, p=0.02, P_A=0.06 for PIU User Iso-Share Curves)

Therefore it is recommended that Internet Service Provider should implement P_R probability in the form of rest-state to improve upon his traffic distribution.

C. BY CIU :

When fig. 8.9 – fig. 8.10 are taken into consideration the final share of operator O_1 is having curve based increasing trend with the variation in opponent blocking L_2. When r_1 is high the final share of O_1 is low for the CIU, but when probability P_R is high along with r_1, final share of operator O_1 is declines constantly. When r_1 and P_R probability both are simultaneously upward operator has to bear the loss of CIU.

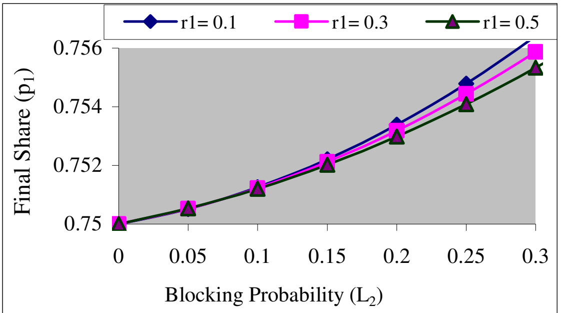


Fig. 8.9 (L_1=0.5, p=0.5, q=0.2, P_{R1}=0.2, P_A=0.3 for CIU User Iso-Share Curves)

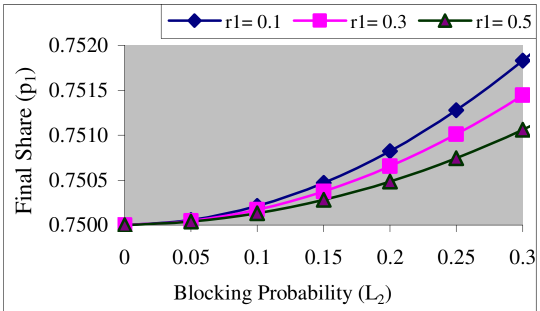


Fig. 8.10 (L_1=0.5, p=0.5, q=0.2, P_{R1}=0.9, P_A=0.3 for CIU User Iso-Share Curves)

By fig. 8.11 and 8.12 it is observed that increasing opponent blocking over P_R probability the final share by the CIU increases. But with the high opponent blocking the self blocking of operator O_1 is also increasing to keep final share of operator improved. So P_R probability is beneficial for increasing the final traffic preparations by the CIU.

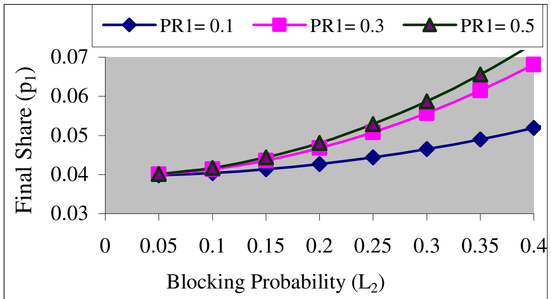


Fig. 8.11 (L_1=0..02, p=0.5, q=0.04, p=0.02, P_A=0.06, r_1=0.02 for CIU User Iso-Share Curves)

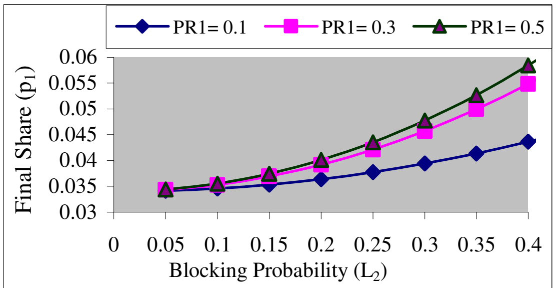


Fig. 8.12 (L_1=0..3, p=0.5, q=0.04, p=0.02, P_A=0.06, r_1=0.02 for CIU User Iso-Share Curves)

9. CONCLUDING REMARKS

The final share loss is of traffic for an operator is found as a linear function of self blocking probability of networks. If the final share goes high then operator of network has to reduce blocking probabilities. The proportion of FU users improves with the increment of r_1 parameter. Moreover P_R and r if both have increment then, faithful user proposition for operator O_1 uplifts. It seems the rest state has strong impact on upliftment of faithful users. To maintain the prefixed final share of PIU, operator O_1 has to reduce his blocking probability in order to keep the earlier initial share. Moreover P_{R1} probability related to rest state if high then operator O_1 has not too much bother about. The CIU users are high affected by opponent network blocking probabilities. They could move to group of FU for high L_2.

10. REFERENCES

[1]. Abuagla Babiker Mohd and Dr. Sulaiman bin Mohd Nor “Towards a Flow-based Internet Traffic Classification for Bandwidth Optimization”, International Journal of Computer Science and Security (IJCSS), 3(2):146-153, 2009

[2]. Ankur Agarwal “System-Level Modeling of a Network-on-Chip”, International Journal of Computer Science and Security (IJCSS), 3(3):154-174, 2009.

[3]. C. Yeian, and J. Lygeres,. "Stabilization of a class of stochastic differential equations with markovian switching, System and Control Letters", issue 9:819-833, 2005.

[4]. D. Shukla, S. Gadewar and R.K. Pathak "A Stochastic model for Space division switiches in Computer Networks", International Journal of Applied Mathematics and Computation, Elsevier Journals, 184(2): 235-269, 2007.

[5]. D. Shukla, Saurabh Jain, Rahul Singhai and R.K. Agarwal "A Markov Chain model for the analysis of Round Robin Scheduling Scheme", International Journal of Advanced Networking and Applications (IJANA), 01(01):01-07, 2009.

[6]. D. Shukla, R.K. Pathak and Sanjay Thakur "Analysis of Internet traffic distribution between two markets using a Markov chain model in computer networks", Proceedings of National Conference on Network Security and Management (NCSM-07), pp. 142-153, 2007.

[7]. D. Shukla and Sanjay Thakur, "Crime based user analysis in Internet traffic sharing under cyber crime", Proceedings of National Conference on Network Security and Management (NCSM-07), pp. 155-165, 2007.

[8]. D. Shukla and Sanjay Thakur, "Rest state analysis in Internet traffic sharing under a Markov chain model", Proceedings of 2nd National Conference on Computer Science & Information Technology, pp. 46-52, 2008.

[9]. Emanual Perzen "Stochastic Processes", Holden-Day, Inc., San Francisco, and California, 1992.

[10]. J. Medhi, "Stochastic Models in queuing theory", Academic Press Professional, Inc., San Diego, CA, 1991.

[11]. J. Medhi, "Stochastic Processes", Ed. 4, Wiley Eastern Limited (Fourth reprint), New Delhi ,1992.

[12]. M. Naldi, "Internet Access Traffic Sharing in A Multi-user Environment", Computer Networks, 38:809-824, 2002.

[13]. M. Newby and R. Dagg, " Optical inspection and maintenance for stochastically deteriorating systems: average cost criteria", Jour. Ind. Stat. Asso., 40(2):169-198, 2002

[14]. Rinkle Aggarwal and Dr. Lakhwinder Kaur "On Reliability Analysis of Fault-tolerant Multistage Interconnection Networks ", International Journal of Computer Science and Security (IJCSS), 2(4):01-08, 2008.

Motion tracking in MRI by Harmonic State Model: Case of heart left ventricle

P. Lionel EVINA EKOMBO evinalio@yahoo.fr
Informatics Statistics & Quality Laboratory
Dhar Mehraz Sciences Faculty
Sidi Mohammed Ben Abdellah University
Fes, Morocco

Mohammed OUMSIS oumsis@fsdmfes.ac.ma
Informatics Statistics & Quality Laboratory
Dhar Mehraz Sciences Faculty
Sidi Mohammed Ben Abdellah University
Fes, Morocco

Mohammed MEKNASSI m.meknassi@gmail.com
Informatics Statistics & Quality Laboratory
Dhar Mehraz Sciences Faculty
Sidi Mohammed Ben Abdellah University
Fes, Morocco

Abstract

We have developed a new method for tracking the closed contour which is based on a harmonic state model (HSM). It tracks the heart's left ventricle (LV) throughout cardiac cycle. This method provides trajectories of points about the contour of the LV, crucial information in cardiac motion analysis. The state vector associated with HSM provides a robust and accurate modeling of contour closed. We rely on the state vector and we use it as local descriptor of region of the LV. This local description enables us to obtain the characteristic points of the contour. Owing the fact that, only light movements between cycle's instants exists. The mapping of these points by the LCSS is relevant. The repetition of this process allows us to build LV trajectories, but also, for further information on its movement, bull eye graphs. The application of the simulation method gives the best results. The same is true on 2D plans sequences extracted from real cine-MRI volume. The trajectories calculated, the generated graphics, allow us to easily observe the difference between a healthy and a sick heart.

Keywords: Harmonic State Model, tracking movement, LCSS, cardiac imagery, MRI.

1. INTRODUCTION

According to the World Health Organization, cardiovascular diseases are a leading cause of death in the world. Thus cardiac function estimation is a main interest. It is useful for the various stages: diagnosis, therapy and intervention. The relevance of the diagnosis is connected to the appreciation of the nature, location and extent of the lesion. The cardiac anomalies affect, according to the cases, the rhythm, the perfusion, the contraction or the metabolism of the heart carried out by the myocardium. It is not only important to detect diseases, but also to evaluate its effect on cardiac structure. This is done by tracking the volumes of cardiac chambers, thickness and movement of the walls of the myocardium. Much of these anomalies are due to a disturbance of the myocardial contractile function, particularly the left ventricle (LV). Tracking of left ventricular movement allows the detection of many cardiovascular diseases. Medical imaging provides essential support for cardiac diagnosis. It provides morphological information (shape, dimensions and volume), dynamic (motion, deformation, speed) and functional (perfusion, metabolic activity). Magnetic Resonance Imaging (MRI), which is non-invasive, allows a complete sight quality of the morphology of the LV. The accuracy of the measurements from the MRI images is demonstrated at [1]. It testifies MRI as the best source for the image analysis of LV. However, extraction of these measures is not helped by the weakness of the resolution and the presence of noise. It can be done either implicitly or explicitly. In the first case, a clinician relying on his job experience can mentally imagine the dynamic behavior of the heart and draw a diagnosis from it. Therefore, one can observe a great variability inter-and intra-observer. In the second case, it is the field of numerical methods through the use of modeling techniques spatiotemporal, excellent modeling alternative, which provides many parameters for the analysis. This employment is done in three stages: segmentation of the LV contour from cardiac images and modeling (2D/3D) the LV pattern, temporal tracking of the LV movements and finally the analysis of calculated parameters in the two previous steps for the issue of diagnosis. In this article a great interest is carried on the second stage of the spatiotemporal modeling, in occurrence the tracking. Tracking the LV movement is a key stage in the process of analysis of cardiac sequences. It provides crucial information on the temporal evolution of LV and at the same time on the state of the heart.

A lot of work has been made, but there are still gaps particularly in terms of accuracy. It also remains costly to implement it in medical software in hospitals. The accuracy of tracking obtained by cardiac MRI-tagged [2, 3] is not matched by these methods. The MRI-tagged does not cover the entire cardiac cycle: it is due to the reabsorbed liquid contrast. This makes it inapt for wide clinical use. The Echocardiography is one of the emerging diagnostic tools for visualizing cardiac structure. It allows diagnosing cardiovascular diseases. Ultrasound imaging is real-time, noninvasive and less expensive than CT and MR. Unfortunately, However, the low signal-to-noise ratio and the multiplicative nature of the noise corrupting the ultrasound images. Then, we have poor object boundaries and make the LV segmentation a difficult task. These images require pretreatment before any use. In [4], authors combine an edge preserving total variation based denoising algorithm and a robust tracker [5]. This process is efficient but stays complex and need a large number of images. In [6], authors use derivatives filters to upgrade grayscale image and it gives good resultants which we could use.

The objective of this speech is to present a new approach relying on both precision and speed in tracking the left ventricle. It is specifically designed to this shape. It is based on a Harmonic State Model (HSM) [7]. This model provides a good modeling of a closed contour (a periodic evolution) and has a robust state vector which exploits the Kalman filter for estimation.

In [7], we proposed a modeling of cardiac motion (left ventricular movement) by a model state harmonic and linear. This motion model combines three essential characteristics of the ventricular movement namely: - access to cardiac dynamics over the whole cycle, - unquestionable robustness to noises, - and direct physical interpretation of functional parameters of LV. The model is linear, periodic and reflects a dynamic model of the decomposition in Fourier series of the cardiac motion. Used as a model state in a Kalman filter, this model offers the advantage of providing an estimate robust to noises, movement parameters such as speed and acceleration which are components of the vector state model. So far, the Harmonic State Model (HSM) has been used in a temporal dimension to model the movement of the LV. The periodicity of the shape of LV (closed surface) also allows us to transpose the model into a spatial dimension. This is enabled with the introduction of shape constraints and smoothness via the harmonic decomposition. This double feature displays the potential of such a model for tracking of 2D/3D wall LV in a sequence of images. We will, therefore, an application of the model HSM in a spatial dimension to model the contours of the LV in order to track local deformations of the regions of LV. In others works, we see that we could use this model in retrieval image system following steps defined in those works [8], [9] and [10].

This paper is divided into four parts. First of all, we will present a state of the art, the HSM model and our tracking method. Then a simulation will made with data to evaluate our approach, followed by an implementation of real patients' data. Finally, we will draw a conclusion and some perspectives.

2. DESCRIPTION OF THE METHOD

2.1 State of art

Many studies have been conducted to provide tools of assistance to the diagnosis. These works range starts from the heart segmentation in image sequences to its analysis, through the modeling of shape and track the movement of the heart. Research on tracking can be divided into two groups depending on the type of movement approach: rigid and not-rigid. The movement of the LV is a not-rigid one. Active contours proposed by Kass and al. [11] ensure this type of tracking and the work arising abound. Nevertheless, they are sensitive to the information. They do not include details related to the LV shape, for example, the periodicity of LV, and the cyclic aspect of cardiac motion. The boundary of active contours is their sensibility to initial positions [12]. Shift methods provide interesting results, but it requires mostly temporal rather complex [13, 14]. The Iterative Closest Point (ICP) presented by Besle and Mc Kay [15] is a method also very much used for tracking. Improved versions are constantly being made; they include the extent of matching (distance measurement, optical flow, etc.) [16], [17], [18]. Others versions are based on the transformation functions (rigid or not-rigid) [18], [19]. We can also mention the algorithms based on optical flow. These optical flow algorithms provide important information for motion analysis of image sequences by attempting to find the vector field which describes spatial movements of image objects over time. Different methods for computation of optical flow can be classified in three main categories: feature-based, gradient-based and correlation-based. [20] gives more details about it. The 2D tracking algorithms are widely presented in [21], [22]. Yilmaz [21] shows that tracking of objects has two phases. The first phase is detection of objects which includes 4 types of algorithms:

- the detection of points that mark points of interest of an image (Harris, KLT, ASG);
- the subtraction of background which constructs a model of representation of the scene;
- segmentation which partitions the image into regions similar (Mean Shift);
- supervised learning that builds a function for mapping data with the expected output (adaptive boosting, SVM).

The second phase, which is the most critical, is the mapping of instances of the object. It includes several families of algorithms:

- filtering that allows the mapping of points of interest from one image to another (particle filters, Kalman filter);
- the tracking body that uses coded information to inside the object region;
- the tracking by kernel research subject compared to a model built in the previous images.

The problem of automatic detection and segmentation of heart volumes have received considerable attentions [23], [24]. In papers [25] and [26], tracking of the left ventricles (LV) have particular attracted interests. It provides clinical significance for specialists to evaluate disease.
Beyond this work, our attention has been drawed by methods which measure distances give goods results; it is specifically the work of Wang [27], Papademetris [28] and Geiger [29].
In this document, we propose a new method to estimate the not-rigid motion that would fill lacks identified in the methods mentioned. In addition to that, it generates the trajectories of the control points of the LV during the cardiac cycle, as well as parameters allowing a classification of the patient as being healthy or pathological. Our method takes into account the periodicity contour of LV and operates through the HSM. It should be noted that the HSM [7] has been used for modeling the temporal movement of the LV (periodic motion). This model is also reducing the impact of noise measures by carrying out a filtering of data through the Kalman filter.

2.2 State Harmonic Decomposition of periodic function

Let *s(t)* be a periodic continuous signal (2D coordinates of a curve for example), *s(t)* harmonic decomposition truncated at level *n* is:

$$s_n(t)=\bar{s}+A_1\,sin(\omega\;t+\varphi_1)+....+A_n\,sin(n\omega\;t+\varphi_n) \tag{1}$$

Where $s_n(t)$ is the truncated Fourier series at order *n*. A_1 … A_n are weighting coefficients, *ω* is the pulsation and φ_1 …φ_n the phases.
Our goal is to find a discrete linear state model according to equation:

$$\begin{cases} S_n(k+1)=F\;\;S_n(k)+\zeta(k) \\ s_n(\,k\,)=H\;\;S_n(k)+\xi(k) \end{cases} \tag{2}$$

where *k* is the discrete variable ($s_n(k)=s_n(k\Delta T)$, *ΔT* the sampling period), $S_n(k)$ is the state vector at index *k*, *F* is the transition matrix, *H* is the measurement matrix and $\zeta(k)$ and $\xi\,(k)$ are zero mean Gaussian noises of covariance matrix respectively *Q(k)* and *R(k)*.

2.2.1 Choice of a state vector

For sample *k+1* corresponding to *t+Δt*, the truncated harmonic decomposition is, according to equation (1):

$$\begin{aligned} s_n(k+1) &= s_n(t+\Delta t) \\ &= \bar{s}+A_1\;\sin\,(\omega\big(t+\Delta t\big)+\varphi_1)+....+A_n\sin\,(n\,\omega\big(t+\Delta t\big)+\varphi_n) \end{aligned} \tag{3}$$

By developing:

$$\begin{aligned} s_n(t+\Delta t) = \bar{s} &+ A_1\sin\,(\omega\;\,t+\varphi_1)\cos\,(\omega\;\,\Delta t)+A_1\cos\,(\omega\;\,t+\varphi_1)\sin\,(\omega\;\,\Delta t)+\cdots \\ &+A_n\sin\,(n\,\omega\;\,t+\varphi_n)\cos\,(n\,\omega\;\,\Delta t)+A_n\cos\,(n\,\omega\;\,t+\varphi_n)\sin\,(n\,\omega\;\,\Delta t) \end{aligned} \tag{4}$$

Our goal is to find a state vector $S_n(t)$ including $s_n(t)$ and its derivative. Assuming $s_n(t)$ a continuous function, infinitely differentiable, lets note $s_n^{(i)}(t)$ the derivatives of $s_n(t)$ at the order *i* with respect to *t*:

$$s_n^{(2(i-1))}(t)=(-1)^{i-1}(\omega)^{2(i-1)}A_1\sin(\omega\,t+\varphi_1)+....+(-1)^{i-1}(n\omega)^{2(i-1)}A_n\sin(n\omega\,t+\varphi_n) \tag{5}$$

All derivatives until order *2(n-1)* give the system of equations:

$$\begin{pmatrix} 1 & \cdots & 1 \\ -\omega^2 & & -(n\omega)^2 \\ \vdots & & \vdots \\ (-1)^{n-1}(n\omega)^{2(n-1)} & \cdots & (-1)^{n-1}(n\omega)^{2(n-1)} \end{pmatrix} \begin{pmatrix} A_1\,sin(\omega t+\varphi_1) \\ \vdots \\ A_n\,sin(n\omega t+\varphi_n) \end{pmatrix} = \begin{pmatrix} s_n(t)-\bar{s} \\ s_n^{(2)}(t) \\ \vdots \\ s_n^{(2(n-1))}(t) \end{pmatrix} \tag{6}$$

The numerical solution of this system gives $A_i\sin(i\omega t+\varphi_i)$ as a linear combination of the elements of the *n+1* dimensional vector $(\bar{s},\;s_n(t),\;s_n^{(2)}(t),s_n^{(4)}(t),\cdots,s_n^{(2(n\text{-}1))}(t))^T$.
Let $\bar{r}_i$ and $r_{i,j}$ *(i = 1,…,n , j = 0,…,n-1)* be the coefficients of this combination:

$$A_i\,\sin\!\big(i\omega t+\varphi_i\big)=\bar{r}_i\bar{s}+\sum_{j=0}^{n-1}r_{i,j}\,s_n^{(2j)}\big(t\big) \tag{7}$$

By deriving equation (7), we obtain the expression of $A_i\cos(i\omega\;t+\varphi_i)$:

$$A_i \cos\left(i\omega t + \varphi_i\right) = \frac{\sum_{j=0}^{n-1} r_{i,j}\, s_n^{\,(2j+1)}(t)}{i\omega} \tag{8}$$

Replacing in the elements $A_i \sin\left(i\omega t + \varphi_i\right)$ and $A_i \cos\left(i\omega t + \varphi_i\right)$ by their expressions (7) and (8), give an expression of *$s_n(t+\Delta t)$* according to the *2n+1* dimensional vector :

$$S_n(t) = \left(\bar{s}, s_n(t), s_n^{(1)}(t), s_n^{(2)}(t), \ldots, s_n^{(2(n-1))}(t), s_n^{(2n-1)}(t)\right)^T \tag{9}$$

Let *a_{2j} (j=1,…,2n+1)* be the coefficients of this linear relation:

$$s_n(k+1) = s_n(t+\Delta t) = a_{2,1}\bar{s} + \sum_{j=2}^{2n+1} a_{2,j} s_n^{(j-2)}(t) = a_{2,1}\bar{s} + \sum_{j=2}^{2n+1} a_{2,j} s_n^{(j-2)}(k) \tag{10}$$

By derivative of equation:

$$\begin{aligned} s_n^{(1)}(t+\Delta t) &= \omega A_1 \cos(\omega\, t + \varphi_1)\cos(\omega\, \Delta t) - \omega A_1 \sin(\omega\, t + \varphi_1)\sin(\omega\, \Delta t) + \cdots \\ &+ n\omega A_n \cos(n\omega\, t + \varphi_n)\cos(n\omega\, \Delta t) - n\omega A_n \sin(n\omega\, t + \varphi_n)\sin(n\omega\, \Delta t) \end{aligned} \tag{11}$$

Replacing in the elements $A_i \sin\left(i\omega t + \varphi_i\right)$ and $A_i \cos\left(i\omega t + \varphi_i\right)$ by their expressions (7) and (8), give an expression of *$s_n^{(1)}(t+\Delta t)$* as a function of vector *$S_n(t)$* like in equation:

$$\begin{aligned} s_n^{(1)}(k+1) &= s_n^{(1)}(t+\Delta t) \\ &= a_{3,1}\bar{s} + \sum_{j=2}^{2n+1} a_{3,j} s_n^{(j-2)}(t) = a_{3,1}\bar{s} + \sum_{j=2}^{2n+1} a_{3,j} s_n^{(j-2)}(k) \end{aligned} \tag{12}$$

The same process can be repeat until $s_n^{(2n-1)}(t+\Delta t)$. Therefore, we build a set of linear relations corresponding to the state model (2).

$$\begin{cases} S_n(k+1) = \begin{pmatrix} \bar{s} \\ s_n(t+\Delta t) \\ s_n^{(1)}(t+\Delta t) \\ \vdots \\ s_n^{(2n-1)}(t+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & & a_{2,2n+1} \\ \vdots & & \ddots & \vdots \\ a_{2n+1,1} & \cdots & \cdots & a_{2n+1,2n+1} \end{pmatrix} \begin{pmatrix} \bar{s} \\ s_n(k) \\ s_n^{(1)}(k) \\ \vdots \\ s_n^{(2n-1)}(k) \end{pmatrix} + \zeta(k) \\ s_n(k) = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \end{pmatrix} S_n(k) + \zeta(k) \end{cases} \tag{13}$$

Note that for the HSM of order n, the proposed state vector is *$S_n(t)$*, dimension *2n+1*, composed by the successive derivative of *$s_n(t)$*.

$$S_n(t) = \left(\bar{s}, s_n(t), s_n^{(1)}(t), \ldots, s_n^{(2(n-1))}(t), s_n^{(2n-1)}(t)\right)^T \tag{14}$$

2.2.2 Computation of the transition matrix

The state equation of the HSM with order (*n+1*) is:

$$S_{n+1}(t+\Delta t) = F_{n+1}\ S_{n+1}(k) + \zeta(k) \tag{15}$$

The transition matrix F_{n+1} can be directly calculated by solving system (6) written at order ($n+1$), but can also be recursively determined from the transition matrix F_n of the harmonics state model of order n. In that case, calculations are simplified. Appendix details the method to compute F_{n+1} recursively from F_n.

2.2.3 Estimation of the state vector component

The estimate of the state vector of $S_n(t)$ the HSM model is obtained by means of a Kalman filter. Kalman filter is a recursive tool to estimate the state vector of a dynamic system from a distribution of observations (noisy measurements) and an evolution equation [30]. Kalman filter is a statistical approach and takes into account the uncertainty associated to the measurements. Using Kalman filter with HSM introduces the constraint of signal periodicity. Kalman filter only supposes the assumption of Gaussian distribution for noises.

The use of such filter is a way to limit the impact of the noise of the measures by realizing a second level of smoothing. The first level of smoothing results from the choice of model order. This both smoothing allows obtaining a modeling shape that conserves more than the characteristic points.

2.3 Point-to-point track

In section 2.2, we have explained all the theory around the construction of the model HSM and its application to a contour of the LV.

The model HSM is a local linear model which is represented by a state vector whose first elements are respectively the average value and the derivatives of the radius of a specific point in the contour at an angle θ. Thanks to the linearity of the model, all state vectors (at a selected order) of the contour may be determined by fixing the step of reconstruction (related to the number of points on the contour to be restored). The study of changes of signs on the matrix of vectors make it possible to find the position of characteristic points which are in a gradual way the extremums (cancellation of the first derivative), inflection points (cancellation of the second derivative), etc.

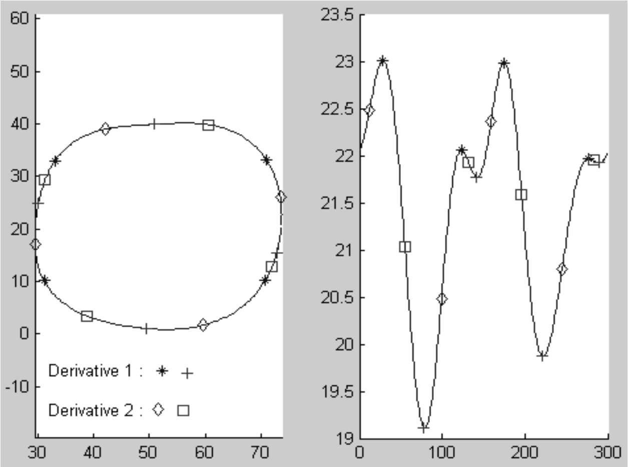


FIGURE 1: The positions of interest point at order 6 for two derivatives (1 and 2).

Thus, between two successive contours of the cardiac cycle which one knows the positions of characteristic points, we can establish a mapping. The risk of being misled by a change in the characteristic points and their positions is minimal given that the movement between two moments of the cardiac cycle is weak.

2.3.1 The mapping

The matching between two contours at successive moments is done in two stages. First of all, the connections are established between the characteristics points of the two contours. It is only after that the control points are connected based on the directions defined in the first step.

2.3.1.1 Mapping characteristic points

Characteristic points or points of interest are organized by groups in order to recognize whether the extremums maximum or minimum, a point of inflection entering or outgoing, etc. This helps to avoid connections between points of interest of different types. The reconstruction of the contour allows obtaining the state vectors of all points of the contour and these vectors are put together to constitute a matrix. The groups of points of interest are obtained through the study of derivatives of state vectors in this matrix. This study is equivalent to detect changes of signs in each column of the matrix. So in the end, we created two great sets of which the first brings together the points of passage of the positive sign to negative sign; and the second set contains all the crossing points of opposite signs. In each set, we have groups that agglomerate points from each column of the matrix. Obviously, the first two columns of the matrix are excluded from this work because they represent the average and the value of the radius, which are constant values. The establishment of groups will help to obtain a framed and fine match.

The application of method LCSS [31],[32],[33],[34] (Longest Common Sub-Sequence) with some constraints allows for the mapping of characteristic points. LCSS method is a dynamic mapping method that improves the DTW (Dynamic Time-Warping) method [35], [36]. It allows you to connect two temporal sequences with different sizes. We use it here, not in the timeframe, but space on sets whose values are not uniformly available in space. Contrary to the DTW, which puts all the points of departure of all in connection with the arrival points, the method LCSS does a match point to point by not establish more than one connection by point and limiting the search in a space defined by a threshold value *δ*.

The expressions defining the method LCSS for a threshold *δ* in a simplified version was made by Bahri [35] in (16).

Let A and B, two sets of values, the method LCSS is defined by the expressions below. In our case, sets of values (A and B) correspond to the indices of points of interest belonging to radius signal.

$$\begin{cases} \mathrm{LCSS}_\delta(\mathrm{A},<>) = \mathrm{LCSS}_\delta(<>,\mathrm{B}) = \mathrm{LCSS}_\delta(<>,<>) = 0 \\ \quad \mathrm{LCSS}_\delta(\mathrm{A},\mathrm{B}) = 1 + \mathrm{LCSS}_\delta(\mathrm{end(A)},\mathrm{end(B)}) \\ \qquad \text{if} \left|\mathrm{last(A)} - last(\mathrm{B})\right| < \delta \\ \quad \mathrm{LCSS}_\delta(\mathrm{A},\mathrm{B}) = \max(\mathrm{LCSS}_\delta(\mathrm{end(A)},\mathrm{B}), \\ \quad \mathrm{LCSS}_\delta(\mathrm{A},\mathrm{end(B)})) \text{ if} \left|\mathrm{last(A)} - last(\mathrm{B})\right| \geq \delta \end{cases} \tag{16}$$

With *last(A)* the last element of sequence A and the rest(A) all elements of A except the last.

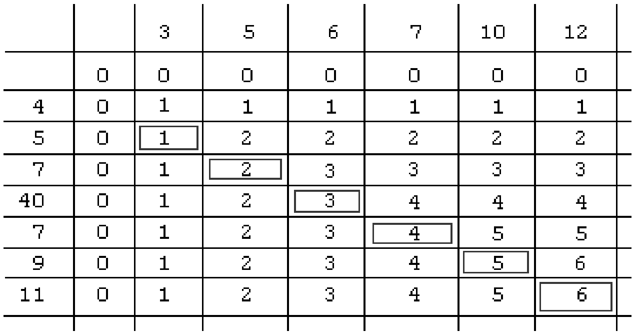
The execution of the method is recursive and LCSS raises the construction of a table. The stages of LCSS for a typified representation table are described in (17). This is the table built to determine the points to put in connection. These points are obtained by determining the optimal path in the table, which way is found of the following specific steps. For two sets *A = {ai} i=1,...,n* and *B = {bj} j=1,...,m,* with n the size of A and m the size of B, we note *T* the table holds values created by the LCSSδ. All cases, *T (i, j)*, which pass through the optimal path puts in connection element *i* of A with element *j* of B. The readings of the path are:

1. Initialization of the trail on the last cell of the table *T (n, m)*;
2. From the case *T (i, j)*, if $\left|a_{i-1} - b_{j-1}\right| < \delta$ then we are going to establish a connection between the element ai of A and bj of B, and also a move to the case *T (i-1, j-1)*.
3. Otherwise, if *T (i, j-1)> = T (i, j-1)* then we move to the case *T (i, j-1)*, otherwise it is to the case *T(i-1, j)*.
4. Return to step *2* until reach case *T (1, .)* or *T (., 1)*.

$$T(i,j) = \begin{cases} 0 & if\ i = 0\, or\ j = 0 \\ 1 + T(i-1, j-1) & if\ \left|a_i - b_j\right| < \delta \\ \max\left(T(i-1,j), T(i,j-1)\right) & if\ \left|a_i - b_j\right| \geq \delta \end{cases} \tag{17}$$

An example of application of LCSS, with and without threshold, will illustrate the method.

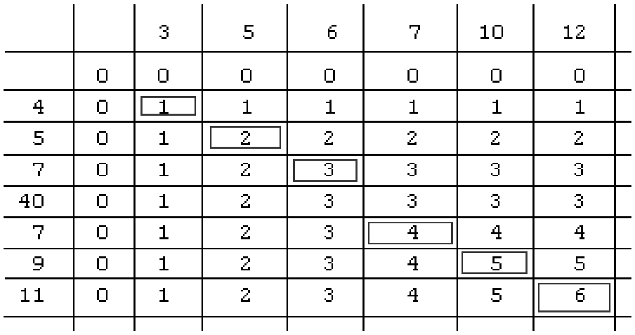
Two sets of values : A={4,5,7,40,7,9,11} and B={3,5,6,7,10,12}
Tables 1 and 2 represent the cumulative values of the LCSS without threshold and with threshold respectively. The boxes are framed on the optimal path:



		3	5	6	7	10	12
	0	0	0	0	0	0	0
4	0	1	1	1	1	1	1
5	0	$\boxed{1}$	2	2	2	2	2
7	0	1	$\boxed{2}$	3	3	3	3
40	0	1	2	$\boxed{3}$	4	4	4
7	0	1	2	3	$\boxed{4}$	5	5
9	0	1	2	3	4	$\boxed{5}$	6
11	0	1	2	3	4	5	$\boxed{6}$

TABLE 1: Table of LCSS without threshold

The optimal path of Table 1 will find that all the points are {5-3, 7-5, 40-6, 7-7, 9-10, 11-12}.



		3	5	6	7	10	12
	0	0	0	0	0	0	0
4	0	$\boxed{1}$	1	1	1	1	1
5	0	1	$\boxed{2}$	2	2	2	2
7	0	1	2	$\boxed{3}$	3	3	3
40	0	1	2	3	3	3	3
7	0	1	2	3	$\boxed{4}$	4	4
9	0	1	2	3	4	$\boxed{5}$	5
11	0	1	2	3	4	5	$\boxed{6}$

TABLE 2: Table of LCSS with threshold set to 10

Table 2, which is obtained by LCSS with value threshold of 10, has a different mapping result with the set of values {4-3, 5-5, 7-6, 7-7, 9-10, 11-12}.

The choice of the threshold value is very delicate. In defining the size of the search window, it plays on possible points of interest that can be mapped.

Once the mapping of intra-group points of interest carried out, we put them all in a single package. The merger of groups reflects a redundancy of information and some differences in the connections. A phase of post-processing is then carried out to overcome these minor problems.

The post-processing is done in three stages. First of all, the ordered fusion points of interest C_t which are matched to those of C_{t+1} is performed (Figure 3.1). This results in the elimination of isolated points of interest. The second step is in amalgamating the close points. This is reflected by the melting points of C_t or C_{t+1} are at a distance less than one parameter set. We preserve only the item on the average of small groups of points to merge (Figure 3.2). The third and final stage involves the cross-connections between points of interest. It includes such points pi and p_j of C_t which are respectively connected to points of p_n and pm of C_{t+1}. This step will restore order in the connections so that we have a liaison between pi and p_n, but also between p_j and pm (Figure 3.3). The post-treatment concluded this section on the matching of characteristic points.

2.3.1.2 Mapping of control points

Once the characteristic points, or at least what remains about it connected, we focus on the control points. It is a set of points positioned uniformly or not, on the outline start C_t and we will track their displacement on contour C_{t+1}. It is not necessary to track all points of the contour, since the control points give an estimate of the movement on the whole contour by the trajectories obtained. For connecting each control point P_i, linear interpolation is performed between the points of the interval containing P_i, defined by two successive characteristic points of the contour C_t, and all points in the interval of their correspondents on contour C_{t+1}. And it only remains to find which points of C_{t+1} corresponds to P_i and the connection is established. We repeat this process for all checkpoints.

2.3.2 Algorithm

The steps involved in the process for tracking a set of control points (P_i) are shown in figure 2.

The process of section 2.3 allows us to obtain trajectories covering the entire cardiac cycle. Figure 3 provides an overview of the matching between two contours.

An additional modeling of these paths is done in order to have filtered trajectories where the effects of noise are eliminated.

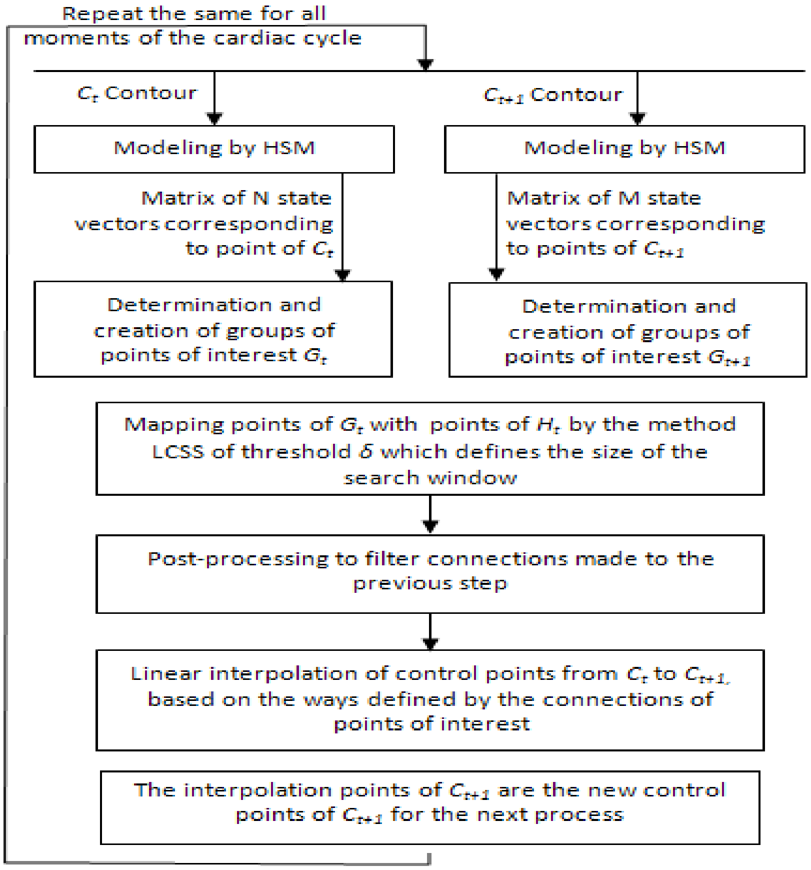

Repeat the same for all moments of the cardiac cycle
C_t Contour
C_{t+1} Contour
Modeling by HSM
Modeling by HSM
Matrix of N state vectors corresponding to point of C_t
Matrix of M state vectors corresponding to points of C_{t+1}
Determination and creation of groups of points of interest G_t
Determination and creation of groups of points of interest G_{t+1}
Mapping points of G_t with points of H_t by the method LCSS of threshold δ which defines the size of the search window
Post-processing to filter connections made to the previous step
Linear interpolation of control points from C_t to C_{t+1}, based on the ways defined by the connections of points of interest
The interpolation points of C_{t+1} are the new control points of C_{t+1} for the next process

FIGURE 2: Process tracking of control point on the entire cardiac cycle

3. EXPERIMENTATION AND RESULTS

Throughout this section, the parameters of the tracking will be evaluated carefully. They are the best settings that will subsequently be implemented for the test phase on the shape of the cardiac cycle LV. Part of these contours is the result of clinical examinations images.

3.1 Parameters and post-processing

The elements involved in this tracking approach are studied and the values of various parameters set to obtain optimal results.

Here, we must provide answers to several questions such as the order of the model to use? What is the best size window of research? What is the criterion value for optimal fusion of characteristic points? and more others questions.

Because the tracking work is done on the contours of LV, which are strongly closed circular contours, a low order model was chosen, in occurrence order *4*. This order is low enough to rebuild the contour in its general shape without any time to take again these small variations introduced by noise measurements and segmentation. This order also provides a consistent, but not excessive, characteristic points for each derivative. At the end, all derivatives are sufficient characteristic points to underline the important characteristics of the shape of contour.

The value threshold *δ* of the LCSS constitutes the size of the window mapping. The tests we have taken to the set at 20. The connection is then established under the best conditions and too great shifts are avoided.

During the post-treatment, a fusion operation of near points of interest is made. The parameter that defines the minimum proximity between two points is set at 4. Figure 3 provides an overview of the evolution of inter-connection points in the process of post-treatment.

In order to ensure a smooth interpolation that occurs just before the matching of control points, and after post-treatment, a generation of 300 points on the contours is necessary. Control points are 20 and are at equal angular distance on the contour of the first cycle.

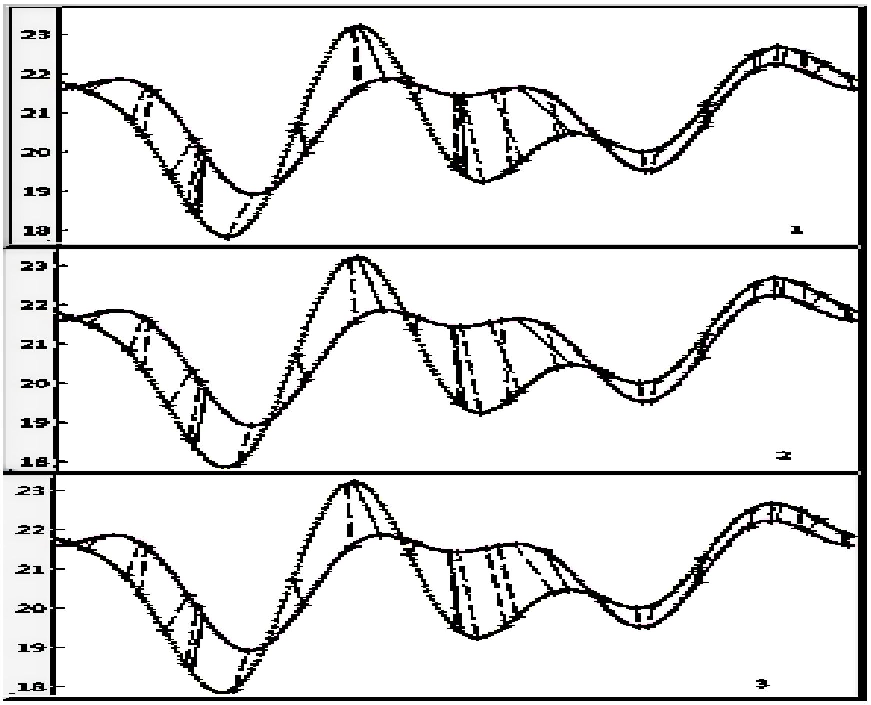


FIGURE 3: Results of the stages of post-treatment, the connectors in dotted lines are for characteristic points and the connectors in continuous lines are for control points.

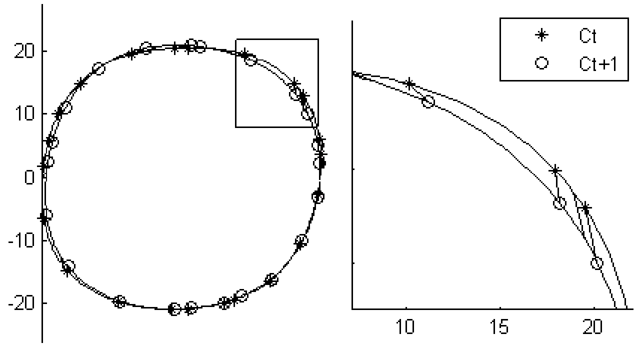


FIGURE 4: Position and correspondence between the interest points of C_t contour and those of C_{t+1} contour

3.2 Real data

The tracking work will be applied to real image sequences that are derived from clinical examinations.

3.2.1 Database

The works on actual data have been made on the basis of 3D+time cine-MRI[1] clinically validated. This database also contains the contours extracted from left ventricle conducted both by experts and by an automatic method [37]. This database is free to use and available to the public through the assistance of the Institut Gaspard-Monge and laboratory A2SI. The images are those of 18 patients with 25 or 22 images 3D per patient, covering the cardiac cycle. Each volume or 3D image contains an independent number of cuts (2D images), which is a function of resolution, hence no consistency between the volumes in this regard. These images, obtained from clinical examinations, are unfortunately not classified whatsoever to distinguish the types of diseases or simply to separate healthy patients from those whom are sick. However, the base remains a good support for the modeling and tracking 2D.

3.2.2 Evaluation of method on one patient

We apply our method for tracking 3D images of patient 4 and, specifically, cut at level 5. We thus have a set of 25 round contours covering the cardiac cycle and are approximately ¼ of the volume of the LV from the apex. Tracking is carried out for 20 points. Figure 6 shows the trajectories obtained for the outer contour (1) and internal (2). The trajectories modeled and filtered of these trajectories are found in Figure 6 (3, 4). We use HSM model to do it.

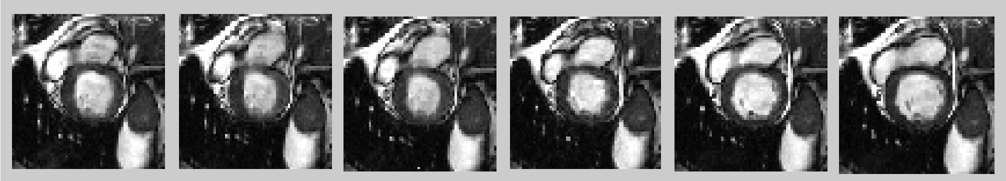


FIGURE 5: Some images of the 3D volumes (cut 5) for the patient 4

[1] 4D heart database : http://www.laurentnajman.org/heart/

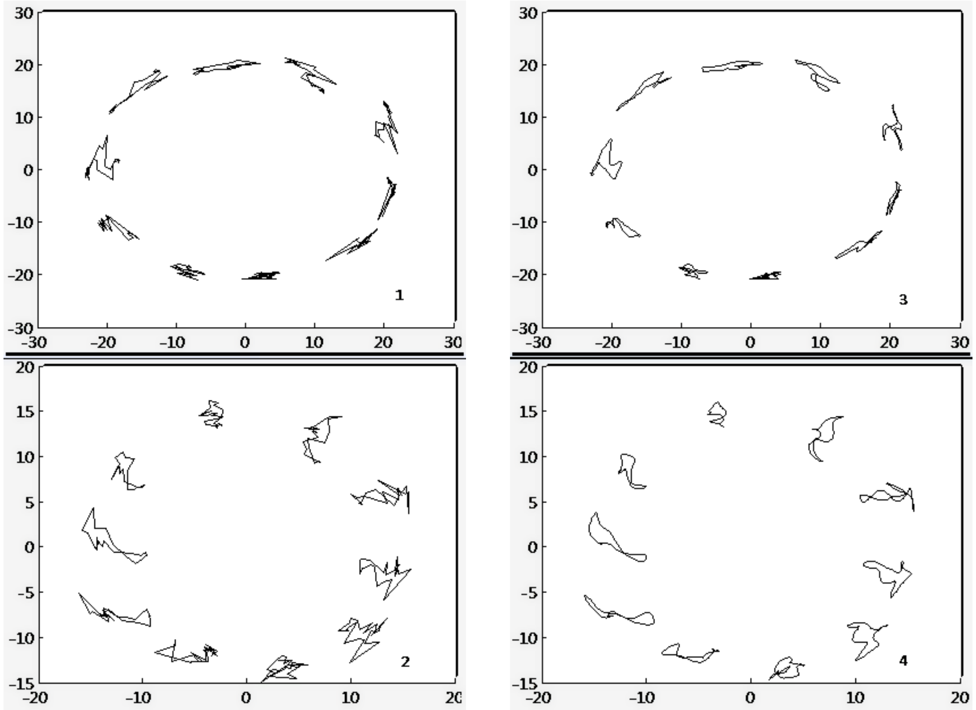


FIGURE 6: Trajectories of control points: (1) epicardium and (2) endocardium; modeled trajectories modeled by HSM model: (1) => (3) and (2) => (4).

We can notice in figure 6 that the trajectories have shapes that are consistent with the cardiac movements. The trajectories of the outer contour are more crushed than the internal contour, and this corresponds to the fact that the outer wall of the LV is more static than the inner wall. The collapse of the trajectories of the inferior part of the inner wall suggests a dysfunction of this part.

3.2.3 Trajectories and analysis of results

Thereafter, the implementation of the method is extended to the whole database. It focuses on both internal and external contours of the left ventricle. It is done by tracking the trajectories of 20 control points evenly distributed on each contour. The resulting information is aggregated into graphs (bull's eyes). These graphs provide the evolution of some parameters in time (throughout the cardiac cycle). The parameters are those which provide a direct interpretation of the movement of the myocardium. On this basis, it is possible to make a diagnosis on the state of the heart. These parameters are:

- The average variation of LV thickness;
- The speed of the endocardium (inner contour);
- The speed of the epicardium (outer contour).

To limit the size of calculations, the LV is divided into three equal parts along the long axis. The short axis cuts corresponding to the middle of each block could be tracked by computing their trajectories. These cuts represent the global movement made by the block where they belong. In the figures that follow, we present the results of three patients each with a different medical state:

- Healthy heart;
- Case reflecting a dyskinetic pathological;
- Case reflecting a hypokinetic pathological.

3.2.3.1 Details of graphs

For each figure, we have three lines that are in the midst of cutting long axis of LV. Each line consists of four graphs that represent the parameters vary during the cardiac cycle. We therefore, respectively (from left to right):

- Trajectories from inner and outer myocardium surfaces;
- The variation of the thickness of the myocardium (for each *t*, this is the difference between the thickness estimated by the model at *t* and the average thickness estimated by the same model);
- The variation of the speed of the inner radius (for each time *t* is the difference between the speed estimated by the model at time *t* and the value average speed estimated by the same model);
- The variation of the speed of the outer radius (for each time *t* is the difference between the speed estimated by the model at time *t* and the average speed estimated by the same model);

The graphs are bull's-eyes of parametric colored by the variations of parameters of each LV 20 regions. The direction of change is from inside to outside.

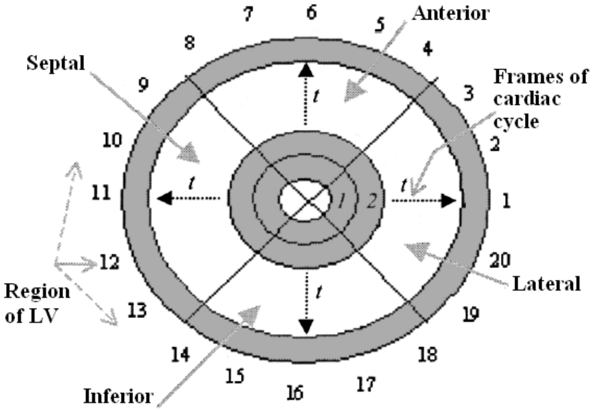


FIGURE 7: The LV is divided into 20 regions then grouped into 4 major regions. Each ring represents a frame cardiac cycle.

3.2.3.2 Analysis of graphs

FIGURE 8:
Graphs showing the variation in thickness have almost a uniform color on each ring (frame). The graphs, on third and fourth columns, show velocity variations remain uniformly circular. Speed variations are regular and not sharp on all regions of the LV. The movement is synchronous in all regions. We are therefore dealing with a heart healthy.

FIGURE 9:
For the three cuts, the variation in thickness of the myocardium is not uniform on the rings. Speed variations are very important with big change from one extreme to another. This applies to both internal and external radius. Then, the heart has a disordered contractile movement. We can conclude that it is a heart with dyskinetic pathology.

FIGURE 10:
For this patient, the trajectories have a small size, almost circular. These trajectories indicate a limited movement of the endocardium and the epicardium. The graphs of thickness variations confirm this observation, with a uniform color. Exceptions for two graphs of speed, these graphs show variations of low amplitudes. All this leads us to believe that we are in the case of a hypokinetic cardiac pathology or weak heart movement.

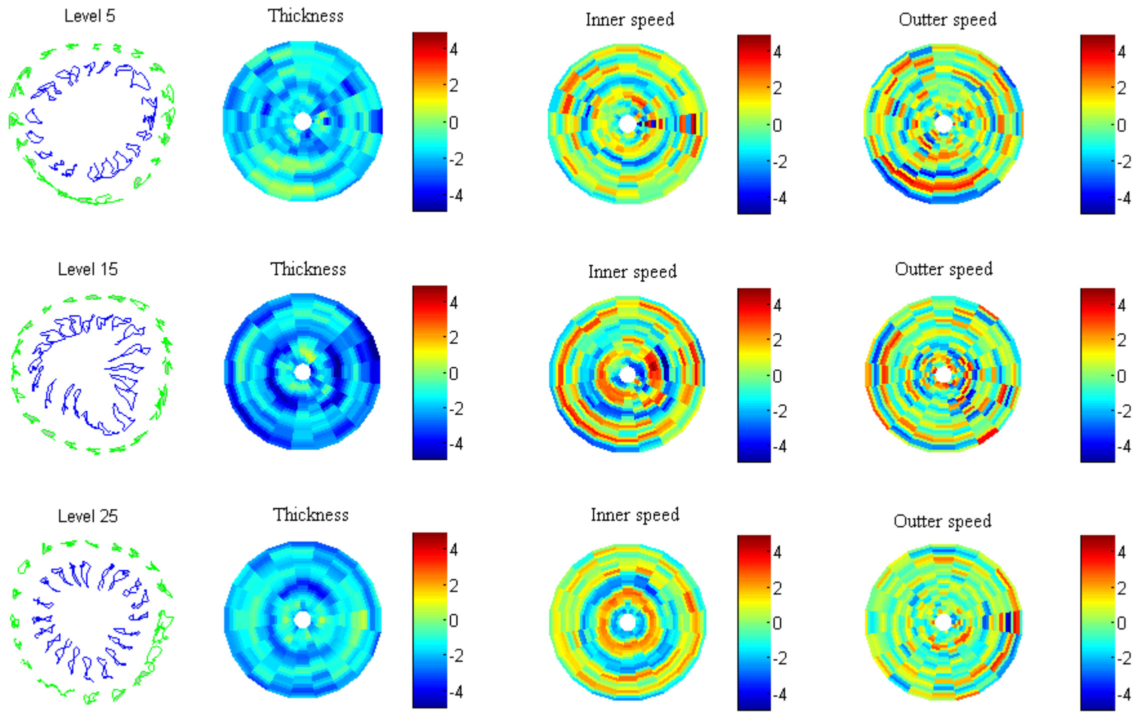


FIGURE 8: Graphs of parameters for a health heart (patient 8)

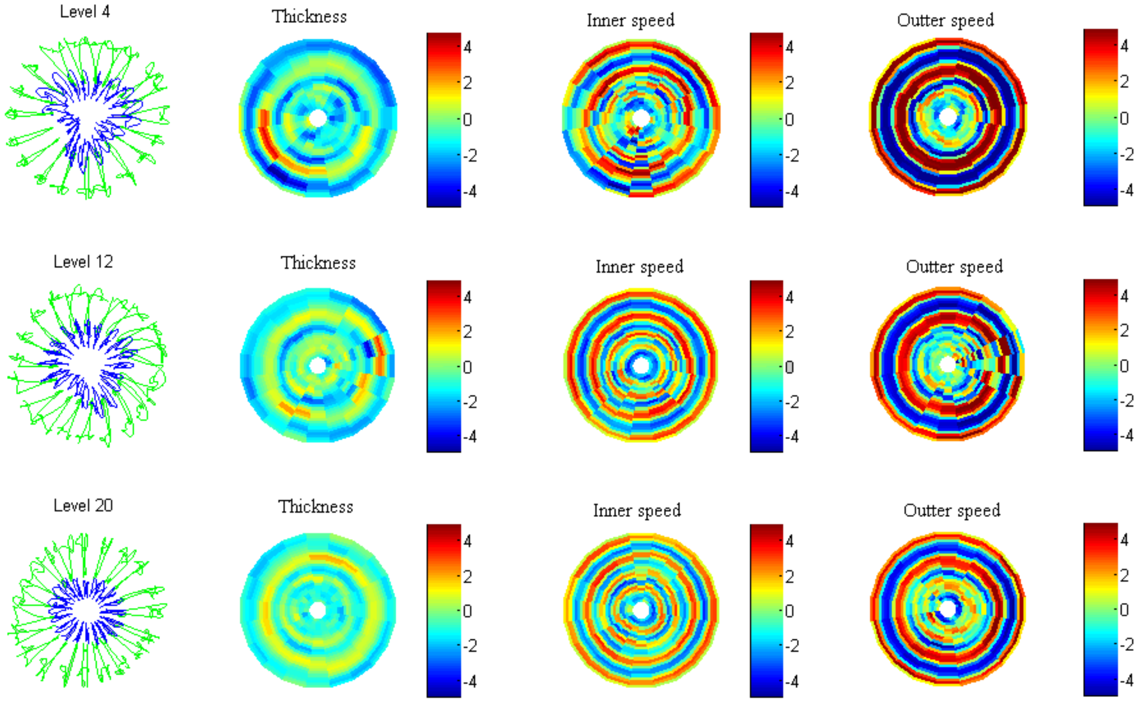


FIGURE 9: Graphs of parameters for a pathologic heart: dyskinetic (patient 6)

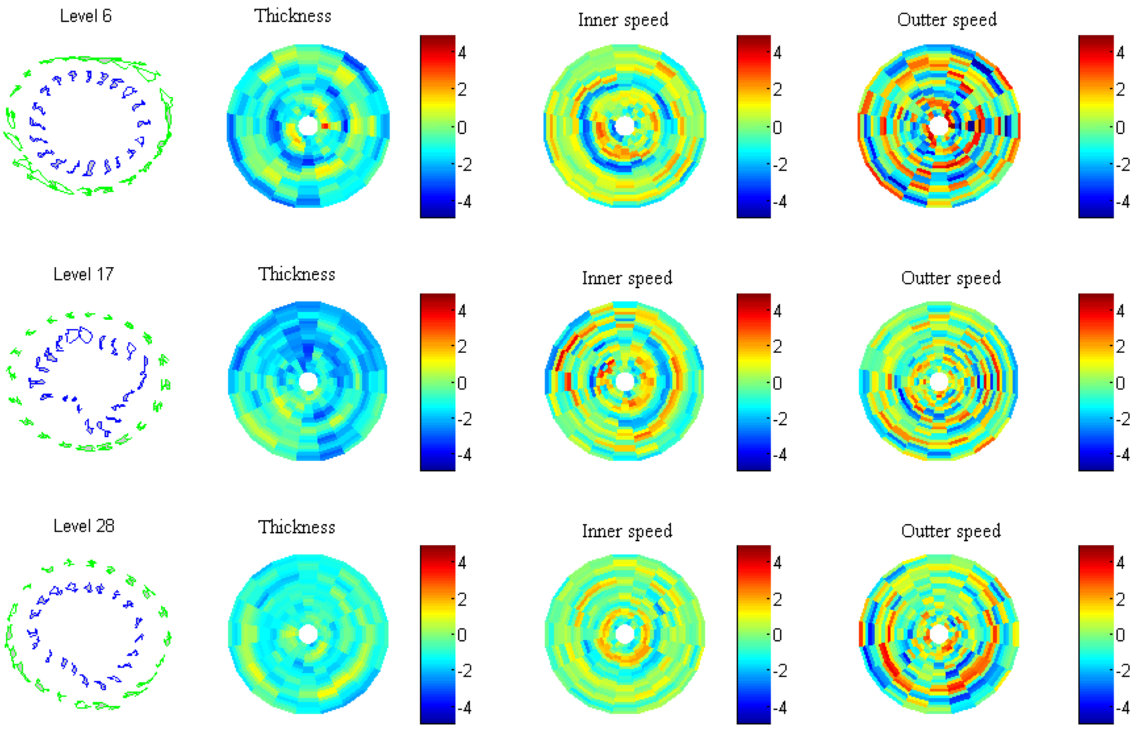


FIGURE 10: Graphs of parameters for a pathologic heart: hypokinetic (patient 13)

4. CONCLUSION & FUTURE WORK

In reviewing the dedicated literature, we have noted many approaches in achieving the modeling and tracking. An important point in tracking the movement of the LV is to make a judicious compromise between the quality of tracking and taking into account the specificities of the latter, as it contains information concerning its pump function. Based on the model of the HSM, we have carried out a 2D contour modeling of LV at each time cycle. Our approach exploits the spatial modeling of the model, which had not yet been made, and its resistance to noise. This model allows finding the characteristic points of the contour. The mapping of these points through the method LCSS and a post-processing allows building trajectories LV points throughout the cardiac cycle. In addition to the trajectories, we gain settings helping with the issue of diagnosis. The experimental results are lead on a base of real pictures provided by clinic exams. The analysis of various parameters generated from these images can provide guidance on the clinical status of patients. All three patients presented in this document are representative of the processing is done on all patients. Our method allows to easily and instantly distinguishing between heart disease and heart healthy through the issue of parameter and the reconstruction of trajectories.

The number of sequences with the actual base is limited and unclassified, we are thinking about moving to a stage of validation on a larger scale. This validation step, we will also make a comparison between our method and the best available technology as the method HARP [38] for MRI tagged.

The result of this work will be done on two main components: first, improving the layout of trajectories, thus improving accuracy, and then, and it is already the case, the application process on envelopes of greater dimensions.

5. REFERENCES

1. P.T. Buser, W. Auffermann, W.W. Holt, S. Wagner, B. Kirker, C. Wolfe, C.B. Higgins. *“Non invasive evaluation of global left ventricular function with use of cine nuclear magnetic resonance”*. Journal of the American College of Cardiology 2(13) pp.1294-1300, 1989.

2. N.F. Osman and J.L. Prince. *“Direct calculation of 2D components of myocardial strain using sinusoidal MR tagging”*. to appear in Proc. of the SPIE Medical Imaging: Image Processing Conference, San Diego, California, 1998.

3. E. McVeigh. *“MRI of myocardial function: motion tracking techniques”*. Magnetic Resonance Imaging, 14(2):137-150, 1996.

4. J. C. Nascimento, João M. Sanches, Jorge S. Marques. *“Tracking the Left Ventricle in Ultrasound Images Based on Total Variation Denoising”*. Proceedings of the 3rd Iberian conference on PRIA, Spain, pp. 628-636, 2007.

5. J. C. Nascimento and J. S. Marques. *“Robust shape tracking in the presence of cluttered background”*. IEEE Trans. Multimedia, vol. 6, no. 6, pp. 852–861, 2004.

6. Chandra Sekhar Panda, Srikanta Patnaik, “*Filtering Corrupted Image and Edge Detection in Restored Grayscale Image Using Derivative Filters*”, International Journal of Image Processing (IJIP), vol.3, issue 3, pp. 105-119, 2009.

7. M. Oumsis, A. D. Sdigui, B. Neyran et I.E. Magnin. *“Modélisation et suivi par modèle d’état harmonique du mouvement ventriculaire gauche du cœur en Imagerie par Résonance Magnétique”*. Dans Traitement du Signal 2000 – volume 17 – n 5/6 – pages 501-516.

8. P. S. Hiremath, Jagadeesh Pujari. “*Content Based Image Retrieval based on Color, Texture and Shape features using image and its complement*”, IJCSS, International journal of computer science and security, vol. 1, issue 4, pp. 25-35, Dec 2007.

9. Rajashree Shettar. “*A Vertical Search Engine: based on Domain Classifier*”, Proceedings of International Journal of Computer Science and Security (IJCSS), vol. 2, issue 4, ISSN (On-line): 1985-1553, pp. 18-27, Nov 2008.

10. P.L. Evina Ekombo, M. Oumsis, M. Meknassi, “*A novel shape descriptor on both local and global curve properties using harmonic state model*”, IRECOS journal, July 2009.

11. M. Kass, A. Witkin, and D. Terzopoulos, *“Snakes Active Contour Models”*. International Journal of Computer Vision, vol. 1, pp. 321-331, 1988.

12. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. *“Active shape models: Their training and application”*. CVIU, 61(1):38–59, 1995.

13. J. Declerck, J. Feldmar, and N. Ayache. *“Definition of a 4D continuous planispheric transformation for the tracking and the analysis of left-ventricle motion”*. Med. Image Anal., vol. 2, no. 2, pp. 197–213, 1998.

14. J. Huang, D. Abendschein, V. Davila-Roman, and A. Amini. *“Spatio-temporal tracking of myocardial deformations with a 4-D B-spline model from tagged MRI”*. IEEE Trans. Med. Imag., vol. 18, no. 10, pp. 957–972, Oct. 1999.

15. P. J. Besl and N. D. McKay. *“A method for registration of 3-D shapes”*. IEEE Transaction on Pattern Analysis and Machine Intelligence, 14(2):239-256, February 1992.

16. J. Declerck, J. Feldmar, N. Ayache. *"Definition of a 4D continuous polar transformation for the tracking and the analysis of LV motion"*. INRIA. N° 3039, November 1996.

17. S. Benayoun, N. Ayache and I. Cohen. *"An adaptive model for 2D and 3D dense non rigid motion computation"*. Technical report 2297, INRIA, May 1994.

18. M. Sühling, M. Arigovindan. *"Myocardial Motion Analysis from B-Mode Echocar-diograms"*. IEEE Transactions on image processing, vol. 4, April 2005.

19. M. J. Ledesma-Carbayo, J. Kybic, M. Desco, A. Santos and M. Unser. *"Cardiac Motion Analysis from Ultrasound Sequences Using Non-rigid Registration"*. MICCAI 2001, pp. 889-896, 2001.

20. Sven Lončarić, Tvrtko Macan. *"Point-constrained optical flow for LV motion detection"*. Proceding SPIE, vol. 3978, 521, 2000.

21. A. Yilmaz, O. Javed, and M. Shah. *"Object tracking: A survey"*. ACM Computer Survey, 38(4):13.1–13.45, 2006.

22. W. Sun, M. Cetin, R. Chan, V. Reddy, G. Holmvang, V. Chandar, A. Willsky. *"Segmenting and Tracking the Left Ventricle by Learning the Dynamics in Cardiac Images"*. Lecture Notes in Computer Science, pp. 553-565 (2005).

23. M. P. Jolly. *"Automatic segmentation of the left ventricles in cardiac MR and CT images."* IJCV, 70(2):151–163, 2006.

24. W. Hong, B. Georgescu, X. S. Zhou, S. Krishnan, Y. Ma, and D. Comaniciu. *"Database-guided simultaneous multi-slice 3D segmentation for volumeric data"*. ECCV, 4:397–409, 2006.

25. M. Dewan, C. H. Lorenz, and G. D. Hager. *"Deformable motion tracking of cardiac structures (DEMOTRACS) for improved MR imaging"*. CVPR, 2007.

26. P. F. U. Gotardo, K. L. Boyer, J. Saltz, and S. V. Raman. *"A new deformable model for boundary tracking in cardiac MRI and its application to the detection of intra-ventricular dyssynchrony"*. CVPR, 1:736–743, 2006.

27. Y. Wang, B. S. Peterson and L. H. Staib. *"Shape-based 3D surface correspondence using geodesics and local geometry"*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 644-651, 2000.

28. X. Papademetris, J. Sunisas, Dione and Duncan. *"Estimation of 3D left ventricular deformation from Echocardiography"*. Medical Image Analysis In Press, March 2001.

29. Geiger, Gupta, Costa, and Vlontzos. *"Dynamic Programming for Detecting, Tracking and Matching Deformable Contours"*. IEEE Transactions PAMI 17(3)294-302, 1995.

30. R.E. Kalman. *"A New Approach to Linear Filtering and Prediction Problems"*. Journal of Basic Engineering, pp. 35-45. March 1960.

31. Tae Sik Han, Seung-Kyu Ko, Jaewoo Kang. *"Efficient Subsequence Matching Using the Longest Common Subsequence with a Dual Match Index"*. 5th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM), Leipzig, Germany, July 18-20, 2007: 585-600.

32. M. Vlachos, G. Kollio, D. Gunopulos. *"Discovering Similar Multidimensional Trajectories"*. 18th IEEE Int. Conf. on Data Engineering (ICDE), pp.673-684, San Jose, USA 2002.

33. Haidar Siba. *"Comparaison des Documents Audiovisuels par Matrice de Similarité"*. PhD thesis, University of Toulouse III - Paul Sabatier, September 2005.

34. Xu Yuxiang; Dafang Zhang; Jiaohua Qin. *"An Improve Algorithm for the Longest Common Subsequence Problem"*. Convergence Information Technology, 2007. International Conference on Volume, Issue, 21-23, pp. 637–639, Nov. 2007.

35. Bahri A., Y. Naïja et G. Jomier. *"Recherche par similarité de séquences dans les bases de données: un état de l'art"*. Manifestation de JEunes Chercheurs STIC (MAJESTIC 2004), Calais, France, Octobre 2004.

36. R. Tavenard, L. Amsaleg, G. Gravier. *"Estimation de similarité entre séquences de descripteurs à l'aide de machines à vecteurs supports"*. Proc. Conf. Base de Données Avancées, Marseille, France, 2007.

37. J. Cousty, L. Najman, M. Couprie, S. Clément-Guimaudeau, T. Goissen, J. Garot. *"Automated, Accurate and Fast Segmentation of 4D Cardiac MR Images"*. Functional Imaging and Modeling of the Heart (2007), pp. 474-483

38. N.F. Osman and J.L. Prince. *"Direct calculation of 2D components of myocardial strain using sinusoidal MR tagging"*. In Proceding of the SPIE Medical Imaging: Image Processing Conference, 1998, San Diego, California.

6. Appendix

Recursive estimation of transition matrix

The Fourier decomposition at order *n* is:

$$s_n(k) = s_n(t) = \bar{s} + A_1 \sin(\omega\, t + \varphi_1) + ... + A_n \sin(n\omega\, t + \varphi_n) \qquad \text{(A.1)}$$

The SHM at the same order is:

$$S_n(k+1) = \begin{pmatrix} \bar{s} \\ s_n(t+\Delta t) \\ s'_n(t+\Delta t) \\ \vdots \\ {s_n}^{(2n-1)}(t+\Delta t) \end{pmatrix} = F_n \begin{pmatrix} \bar{s} \\ s_n(k) \\ s'_n(k) \\ \\ {s_n}^{(2n-1)}(k) \end{pmatrix} + \zeta(k) \qquad \text{(A.2)}$$

With

$$F_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & & a_{2,2n+1} \\ \vdots & & \ddots & \vdots \\ a_{2n+1,1} & \cdots & \cdots & a_{2n+1,2n+1} \end{pmatrix} \qquad \text{(A.3)}$$

The Fourier decomposition at order *n+1* is:

$$s_{n+1}(t) = s_n(t) + s^*{}_{n+1}(t) \qquad \text{(A.4)}$$

With

$$s^{*}{}_{n+1}(t) = A_{n+1}\, sin(\,(\,n+1\,)\omega\; t+\varphi_{n+1}\,) \tag{A.5}$$

A state form of this harmonic is given by:

$$\begin{aligned}\begin{pmatrix} s^{*}{}_{n+1}(t+\Delta t) \\ s^{*'}{}_{n+1}(t+\Delta t)\end{pmatrix} &= \begin{pmatrix} \cos(\,(n+1)\omega\,\Delta t) & \dfrac{\sin(\,(n+1)\omega\,\Delta t)}{(n+1)\omega} \\ -(n+1)\omega\sin(i\omega\,\Delta t) & \cos(\,(n+1)\omega\,\Delta t)\end{pmatrix}\begin{pmatrix} s^{*}{}_{n+1}(t) \\ s^{*'}{}_{n+1}(t)\end{pmatrix} \\ &= \alpha_i \begin{pmatrix} s^{*}{}_{n+1}(t) \\ s^{*'}{}_{n+1}(t)\end{pmatrix}\end{aligned} \tag{A.6}$$

We want to obtain the SHM at order (*n+1*):

$$\begin{pmatrix} \bar{s} \\ s_{n+1}(k+1\,) \\ s_{n+1}{}^{(\,1\,)}(k+1\,) \\ \vdots \\ s_{n+1}{}^{(\,2n+1\,)}(k+1\,)\end{pmatrix} = F_{n+1}\begin{pmatrix} \bar{s} \\ s_{n+1}(k) \\ s_{n+1}{}^{(\,1\,)}(k) \\ \\ s_{n+1}{}^{(\,2n+1\,)}(k)\end{pmatrix} + \zeta(k) \tag{A.7}$$

Equation (6) written at order *n+1* gives:

$$\begin{pmatrix} 1 & \cdots & 1 \\ -\omega^2 & & -((n+1)\omega)^2 \\ \vdots & & \vdots \\ (-1)^n & \cdots & (-1)^n(n\omega)^{2(n)}\end{pmatrix}\begin{pmatrix} A_1\, sin(\omega t+\varphi_1\,) \\ \vdots \\ A_{n+1}\, sin(\,(n+1)\omega t+\varphi_{n+1}\,)\end{pmatrix} = \begin{pmatrix} s_{n+1}(t)-\bar{s} \\ s_{n+1}{}^{(\,2\,)}(t) \\ \vdots \\ s_{n+1}{}^{(\,2(n))}(t)\end{pmatrix} \tag{A.8}$$

By mean of upper triangularization (numerical solution), the last component of the harmonic decomposition is:

$$\begin{aligned} s^{*}{}_{n+1}(t) &= A_{n+1}\sin(\,(n+1)\omega\, t+\varphi_{n+1}\,) \\ &= \bar{r}_{n+1}\bar{s} \;+\sum_{j=0}^{n} r_{n+1,j}\, s_{n+1}{}^{(\,2\,j)}(t)\end{aligned} \tag{A.9}$$

By derivation:

$$s^{*\prime}{}_{n+1}(t) = \sum_{j=0}^{n} r_{n+1,j}\, s_{n+1}{}^{(\,2\,j+1\,)}(t) \tag{A.10}$$

From Equations (A.4) and (A.9):

$$\begin{aligned} s_n(t) &= s_{n+1}(t) - s^{*}{}_{n+1}(t) \\ &= s_{n+1}(t) - \bar{r}_{n+1}\bar{s} \;-\sum_{j=0}^{n} r_{n+1,j}\, s_{n+1}{}^{(\,2\,j)}(t)\end{aligned} \tag{A.11}$$

And derivation of order *2i*:

$$s_n^{(2i)}(t) = s_{n+1}^{(2i)}(t) - (-1)^i\left((n+1)\omega\right)^{2i} A_{n+1}\sin(\,(n+1)\omega\, t+\varphi_{n+1}\,) \tag{A.12}$$

$$s_n^{(2i)}(t) = s_{n+1}^{(2i)}(t) - (-1)^i \left((n+1)\omega\right)^{2i} \left[\bar{r}_{n+1}\bar{s} + \sum_{j=0}^{n} r_{n+1,j}\, s_{n+1}{}^{(2j)}(t) \right] \tag{A.13}$$

By derivation

$$s_n^{(2i+1)}(t) = s_{n+1}^{(2i+1)}(t) - (-1)^i \left((n+1)\omega\right)^{2i} \left[\sum_{j=0}^{n} r_{n+1,j}\, s_{n+1}{}^{(2j+1)}(t) \right] \tag{A.14}$$

By writing Equation (A.4) at $t + \Delta t$

$$s_{n+1}(t+\Delta t) = s_n(t+\Delta t) + s^{*}{}_{n+1}(t+\Delta t) \tag{A.15}$$

HSM at order *n* gives:

$$s_n(t+\Delta t) = a_{2,1}\bar{s} + \sum_{j=0}^{2n-1} a_{2,j+2}\, s_n{}^{(j)}(t) \tag{A.16}$$

Equation (A.6) gives

$$\begin{aligned} s^{*}{}_{n+1}(t+\Delta t) &= \cos((n+1)\omega\,\Delta t)\, s^{*}{}_{n+1}(t) \\ &+ \frac{\sin((n+1)\omega\,\Delta t)}{(n+1)\omega}\, s^{*'}{}_{n+1}(t) \end{aligned} \tag{A.17}$$

Replacing in equation (A.15):

$$\begin{aligned} s_{n+1}(t+\Delta t) &= a_{2,1}\bar{s} + \sum_{j=0}^{2n-1} a_{2,j+2}\, s_n{}^{(j)}(t) \\ &+ \cos((n+1)\omega\,\Delta t)\, s^{*}{}_{n+1}(t) + \frac{\sin((n+1)\omega\,\Delta t)}{(n+1)\omega}\, s^{*'}{}_{n+1}(t) \end{aligned} \tag{A.18}$$

Replacing $s_n{}^{(j)}$ (A.13, A.14), $s^{*}{}_{n+1}(t)$ (A.5) and $s^{*'}{}_{n+1}(t)$ (A.10) by their expression, we obtain equation (A.19):

$$\begin{aligned} \mathrm{s}_{n+1}(t+\Delta t) = &\left(a_{2,1} + \bar{r}_{n+1} \left(\cos((n+1)\omega\,\Delta t) + \sum_{i=1}^{n} a_{2,2i} \left((-1)^{(i)} \left((n+1)\omega\right)^{2(i-1)} \right) \right) \right) \bar{s} + \\ &+ \sum_{j=0}^{n-1} \left(a_{2,2(j+1)} + \cos((n+1)\omega\,\Delta t)\, r_{n+1,j} + r_{n+1,j} \sum_{i=1}^{n} a_{2,2i} \left((-1)^{(i)} \left((n+1)\omega\right)^{2(i-1)} \right) \right) s_{n+1}{}^{(2j)}(t) \\ &+ \sum_{j=0}^{n-1} \left(a_{2,2(j+1)+1} + \frac{\sin((n+1)\omega\,\Delta t)}{(n+1)\omega} r_{n+1,j} + r_{n+1,j} \sum_{i=1}^{n} a_{2,2i+1} \left((-1)^{(i)} \left((n+1)\omega\right)^{2(i-1)} \right) \right) s_{n+1}{}^{(2j+1)}(t) \\ &+ \left(a_{2,2(n+1)} + \cos((n+1)\omega\,\Delta t)\, r_{n+1,n} + r_{n+1,n} \sum_{i=1}^{n} a_{2,2i} \left((-1)^{(i)} \left((n+1)\omega\right)^{2(i-1)} \right) \right) s_{n+1}{}^{(2n)}(t) \\ &+ \left(a_{2,2(n+1)+1} + \frac{\sin((n+1)\omega\,\Delta t)}{(n+1)\omega} r_{n+1,n} + r_{n+1,j} \sum_{i=1}^{n} a_{2,2i+1} \left((-1)^{(i)} \left((n+1)\omega\right)^{2(i-1)} \right) \right) s_{n+1}{}^{(2n+1)}(t) \end{aligned} \tag{A.19}$$

This expression gives the second line of the matrix *F_{n+1}*. The expressions $s_{n+1}^{(i)}(t+\Delta t)$ are calculated by derivation of equation (A.19) with respect to *Δt*. In conclusion we have determined the coefficients connecting the vector $s_{n+1}^{(i)}(t+\Delta t)$ to the vector $s_{n+1}^{(i)}(t)$. These coefficients constitute the elements of the transition matrix *F_{n+1}*.

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA