Towards a Hierarchical Temporal Memory Based Self-managed Dynamic Trust Replication Mechanism in Cognitive Mobile Ad-Hoc Networks

RICARDO J. RODRIGUEZ
Raytheon – Network Centric Systems
8333 Bryan Dairy Road
Largo, FL 33777
UNITED STATES OF AMERICA
Ricardo_J_Rodriguez@raytheon.com

JAMES A. CANNADY, Ph.D.
Graduate School of Computer and Information Sciences
Nova Southeastern University
Fort Lauderdale, Florida
UNITED STATES OF AMERICA
cannady@nova.edu

Abstract: — Dynamic trust models and replication approaches that adequately address the security needs of wireless cognitive ad hoc networks (MANET) have proven quite difficult to develop. The inherent decentralized nature of ad hoc networks and their collaborative services give rise to a series of vulnerabilities that can be exploited by malicious entities. This problem is exacerbated by the fact that risk assessment models show high levels of complexity, which suggest significant domain constraints and lack of true human-like reasoning. Recent work in the area of HTM demonstrates the ability to mimic higher level cognitive skills. This paper discusses the potential of using HTM to improve human-like reasoning in the replication of trust information in cognitive MANETs.

Key-Words: — MANETs, Trust, Threats, HTM, Hierarchical Temporal Memory, Vulnerabilities, Risk Assessment

1 INTRODUCTTION

From a human perspective, trust is a very subjective measure. There are numerous elements or variables used to represent its value. As suggested by D' Arienzo [2], the ability of a system to adapt to ever changing requirements without human intervention is a clear challenge that needs to be resolved. This lack of dynamic adaptation can also be extended to existing trust models and in particular to trust data distribution methods [1] [12] [16], which are responsible for the accurate and timely dissemination of trust changes in order to protect the entire network. Current models acknowledge attributes changing their values, but they do not acknowledge the implication of inaccurate trust data distribution.

This inability places significant restrictions on ad-hoc networks, in particular cognitive networks, to only allow for scenarios where a compromised device can successfully detect malicious behavior and isolate itself. The fact that a number of attacks utilize one or more compromised devices in the target network to further the attack clearly demonstrates that this standard approach is not enough. For example, the decision of a secondary device in an ad hoc network to establish a connection with another initiating secondary can be made by evaluating trust values [12]. If a malicious device is trying to gain access in order to hijack another device and the attributes used to determine the trust value are compromised, the secondary device and ultimately the entire network can be compromised.

Trust evaluations can be enhanced by leveraging automated risk assessment techniques. Novel approaches to accomplish have been proposed and studied in recent years. They leverage research across multiple disciplines, including, but not limited to, artificial intelligence, game theory, knowledge management, and others. However, existing risk assessment models are limited to predetermined set of mathematical models and functions, and, although some of them attempt to leverage artificial intelligence (e.g. neural networks), their capabilities are constrained to specific domains and limited by the lack of temporal awareness.

In 1985, Jeff Hawkins stumbled on the gap that exists between neuroscience and AI. In 2004, he published a book, titled "On Intelligence"[6], where he describes in detail his theory of higher level thought in the neocortex. The concepts of his theory are not new, but the order and sequence is. His approach has propelled additional research to develop human-like cognitive capabilities in systems. As noted by "Business Week" magazine, this theory is based on the premise that intelligence is rooted in the brain's ability to access memories rather that in its ability to process new data.

Hawkins founded NUMENTA, a research company that created NuPIC (Numenta's Platform for Intelligent Computing), which supports the creation of hierarchical temporal memory models. The company's licensing approach to its technology, including intellectual property rights, and resources further incentivize research in this area by both industry and academia.

2 Previous Work

2.1 Automated Risk Assessment

Ke He, Zhiyong Feng, and Xiaohong Li (2008) proposed a novel approach to software security risk assessment during the design stage [7]. The approach relies on attacks and a model to depict system functions, assets, actors, and threats. It also leverages the concept of trust level and trust boundaries. The model is decomposed into sub-nodes merged utilizing AND logic. It aims to identify intentions and goals by dividing the model into three high-level stages: creation and validation of attack scenarios, software security testing according to the attacks, and threat mitigation.

After describing each one of the stages, including a complex attack generation algorithm, the researchers evaluated the model in a simulated online banking system. They concluded that attack scenarios can bridge the gap between system functions design and software security analysis via attack links and mitigation links, which can ultimately contribute to security risk reduction as well as cost reduction.

Wei He, Chunhe Xia, Haiquan Wang, and Cheng Zhang, Yi Ji (2008) presented a model to quantify the threat probability in network security risk assessments [9]. Most of the methods tend to consider the attacker and defender separately. In their work, the attacker and defender are considered game players. Utilizing game theory, the behaviors of the attacker are predicted.

The researchers described the risk assessment framework, which is comprised of network status information collection, attack-defense knowledge library, game theoretical model, risk computation model, and system risk. This game theoretical attack-defense model (GTADM) consists of a mathematical model, a series of definitions, cost benefit analysis of attacker and defender, and equilibrium.

The researchers illustrated the application of GTADM in a demilitarized zone (DMZ) containing multiple network devices providing different services. The results of the simulation established the risk probabilities for each network device as well as the overall network as a unit. They finalized by stating that the weights of the nodes as well as the incidence relationship between the threats were not considered and should be part of the next steps in their research.

These same researchers also proposed a game theory based model for performing risk assessments [8]. The model analyzes the relationship between processes and entities, including network, knowledge, asset, vulnerability, threat, control, impact, probability, and risk.

The researchers described the risk assessment concepts such as asset identification, control identification, and vulnerability identification, among others. They explained the role of game theory in network security risk assessments, which is the foundation for building the proposed framework. The main high level components of this model include a data collection layer, information refinement layer, and risk computing layer. They evaluated the model, which showed promising results by showing the calculated risk using a simulation. However, they acknowledged the need to further enhance the model and test it in a real network.

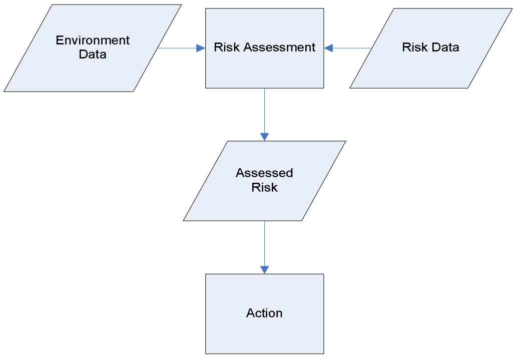


Figure 1. High level representation of Risk Assessment Process

Figure 1 describes the overall goal of automated risk assessments. By leveraging knowledge of both existing risk data and the environment, an assessment is performed. The result of the assessment, the assessed risk, is then utilized to determine a next step or action.

2.2 HTM Based Automated Risk Assessment

Rodriguez and Cannady (2010) proposed a new approach to enhance the current state of the art of automated risk assessments by leveraging HTMs [18].

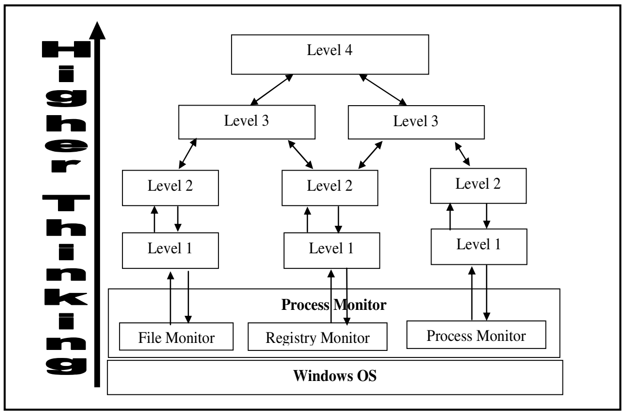
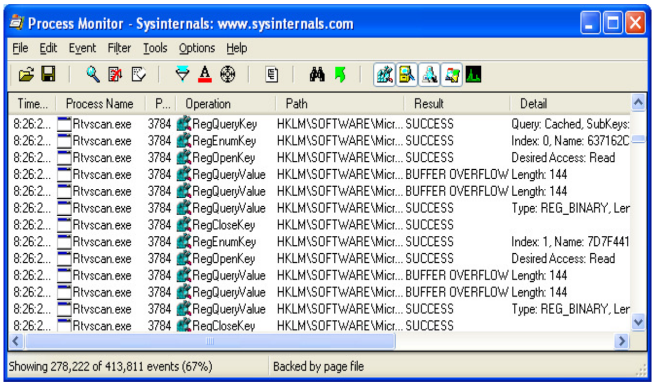


Figure 2. Hierarchical Temporal Memory Model for Automated Risk Assessment in MS Windows Hosts

Figure 2 depicts the Windows environment sensory hierarchy used as a starting point. As levels go up, higher level thinking occurs. At the lower levels, detailed data gathering and analysis is performed. By using MS process monitor (see figure 3), files, registry, and process activities are gathered (i.e. sensory data). This data is then processed at the higher levels as well as passed down to each sensor through feedback lines. For example, data gathered by the registry monitor could be passed all the way to level 3 and then back down to the process monitor.



Time...	Process Name	P...	Operation	Path	Result	Detail
8:26:2...	Rtvscan.exe	3784	RegQueryKey	HKLM\SOFTWARE\Micr...	SUCCESS	Query: Cached, SubKeys:
8:26:2...	Rtvscan.exe	3784	RegEnumKey	HKLM\SOFTWARE\Micr...	SUCCESS	Index: 0, Name: 637162C
8:26:2...	Rtvscan.exe	3784	RegOpenKey	HKLM\SOFTWARE\Micr...	SUCCESS	Desired Access: Read
8:26:2...	Rtvscan.exe	3784	RegQueryValue	HKLM\SOFTWARE\Micr...	BUFFER OVERFLOW	Length: 144
8:26:2...	Rtvscan.exe	3784	RegQueryValue	HKLM\SOFTWARE\Micr...	BUFFER OVERFLOW	Length: 144
8:26:2...	Rtvscan.exe	3784	RegQueryValue	HKLM\SOFTWARE\Micr...	SUCCESS	Type: REG_BINARY, Ler
8:26:2...	Rtvscan.exe	3784	RegCloseKey	HKLM\SOFTWARE\Micr...	SUCCESS	
8:26:2...	Rtvscan.exe	3784	RegEnumKey	HKLM\SOFTWARE\Micr...	SUCCESS	Index: 1, Name: 7D7F441
8:26:2...	Rtvscan.exe	3784	RegOpenKey	HKLM\SOFTWARE\Micr...	SUCCESS	Desired Access: Read
8:26:2...	Rtvscan.exe	3784	RegQueryValue	HKLM\SOFTWARE\Micr...	BUFFER OVERFLOW	Length: 144
8:26:2...	Rtvscan.exe	3784	RegQueryValue	HKLM\SOFTWARE\Micr...	BUFFER OVERFLOW	Length: 144
8:26:2...	Rtvscan.exe	3784	RegQueryValue	HKLM\SOFTWARE\Micr...	SUCCESS	Type: REG_BINARY, Ler
8:26:2...	Rtvscan.exe	3784	RegCloseKey	HKLM\SOFTWARE\Micr...	SUCCESS	

Showing 278,222 of 413,811 events (67%)   Backed by page file

Figure 3. MS Windows Process Monitor

A series of small programs were coded, executed, and tracked using process monitor. These programs contained a series of vulnerabilities, including the following:

- Heap overflow
- Stack overflow

Figure 4 shows a well known code snippet used to show a stack overflow attack [25]. A total of sixty-four similar code snippets (e.g. changing sequence of steps, changing strings, etc..) with the potential for heap and stack overflows were used to capture process monitor data. The data was then gathered and analyzed to represent potential values of the HTM nodes. Fifty-two code instances were used for training and twelve for testing. Of course, as additional runs were executed, more data was gathered from the process monitor and added back to the network for training.

```
#define BUFSIZE 16
int main(int argc, _char* argv[])
{
  char *buffer1 = (char *)malloc(BUFSIZE);
  if (buffer1 == NULL)
    return 0
  char *buffer2 = (char *)malloc(BUFSIZE);
  if (buffer2 == NULL)
    return 0
  memset(buffer1, 'A', BUFSIZE-1);
  buffer1[BUFSIZE-1] = '\0';
  memset(buffer2, 'A', BUFSIZE-1);
  buffer2[BUFSIZE-1] = '\0';
  printf("buffer1 pointer = %p, buffer2 pointer = %p",   buffer1, buffer2);
  printf("\n\nValue in buffer1: %s", buffer1);
  printf("\nValue in buffer2: %s", buffer2);
  printf("\n\nEnter new value to be placed in buffer1 with gets(): ");
  gets(buffer1);
  printf("buffer1 pointer = %p, buffer2 pointer = %p", buffer1, buffer2);
  printf("\n\nValue in buffer1: %s", buffer1);
  printf("\nValue in buffer2: %s", buffer2);
  printf("\n\nPress enter to (try) and free buffer1...");
  getchar();
  free(buffer1);
  printf("\n\nPress enter to (try) and free buffer2...");
  getchar();
  free(buffer2);
  return 0;
}
```

Figure 4. Stack Overflow code snippet.

A simple data aggregation process and a binary representation method were used to represent the output from the process monitor. Table 1 shows the binary bits associated with each operation detected by process monitor.

Operation	First Bit	Second Bit	Third Bit
Read File	0	0	1
Close File	0	1	0
Create File	0	1	1
Reg Open	1	0	0
Reg Read	1	0	1
Reg Query	1	1	0

Table 1. Binary representation of process monitor detected operations

After the first three bits, an additional 253 bits were used to capture the last section of the path information (e.g. section after last special character) and the result (e.g. SUCCESS, FILENOTFOUND, BUFFEROVERFLOW, etc..) as shown by process monitor. The use of the last section of the path information resulted from the fact that in order to capture all the path information, a larger block size would be required, which was not practical for the HP 2.0 GHz computer and 2 GB RAM used for the test. To capture the path, standard ASCII code was used (i.e. 8-bit per character). Figure 5 demonstrates an example output from process monitor used to train the HTM network.

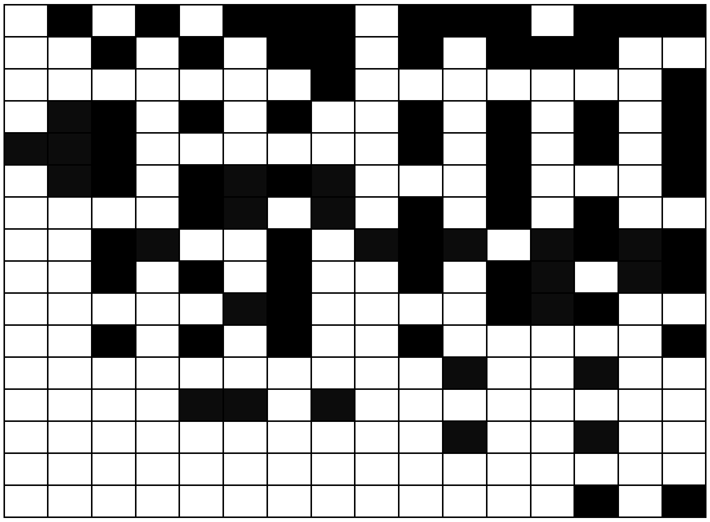


Figure 5. Binary representation of process monitor output

The HTM network was implemented using NuPIC. An initial network based on Numenta's bitworms example code was used as the foundation (see figure 6). Additional sample projects, such as net_construction, which clearly demonstrate the creation of multiple node types, were utilized as well. This was extended to further accommodate the large number of bits required to represent each sequence of process monitor detected behaviors.

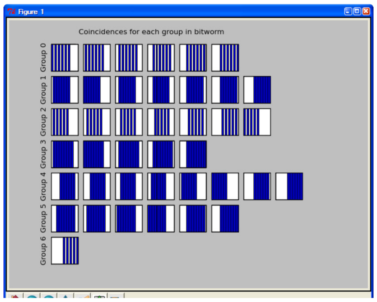


Figure 6. Bitworm Example

The work performed through multiple runs resulted in the determination of nodes and their interconnections. As seen in figure 7, after hundreds of runs, the network eventually converged and was able to identify potential vulnerabilities 37.5% of the time. A key observation during the test is that the separation between training and test accuracy diminished as the number of runs increased.

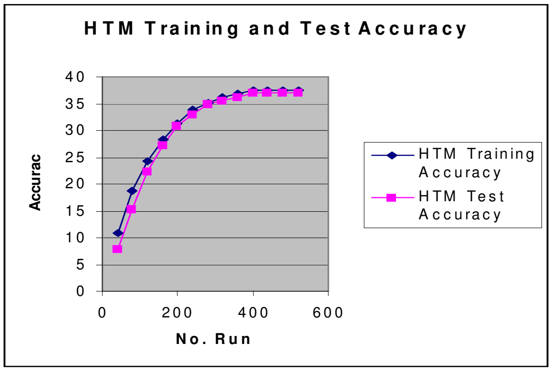


Figure 7. HTM training and test accuracy

Although the results above are not spectacular, they are promising. It is obvious that further research is required since the amount of exploits is substantial and always growing, this research was limited to two types of well known vulnerabilities, the amount of path information used from MS processes monitor was minimal, and the type of connections and number nodes can be dramatically changed.

2.3 Dynamic Trust Replication

Different approaches have been proposed to address the need to cope with dynamic malicious behaviors, most notably peer-to-peer dynamic trust metrics. These methods prove valuable in measuring trust over time in a one-on-one interaction. As suggested by Chang in his paper "A Dynamic Trust Metric for P2P systems" [1], additional work is required to ensure truthful feedback is provided. This flaw can easily be exploited by malicious users who, by leveraging passive and false node attacks, could compromise the trust level data.

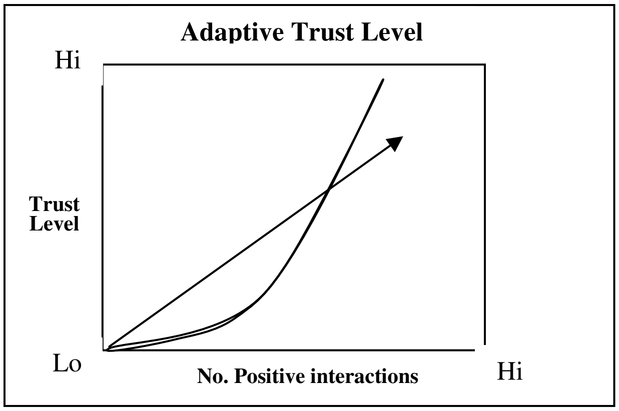
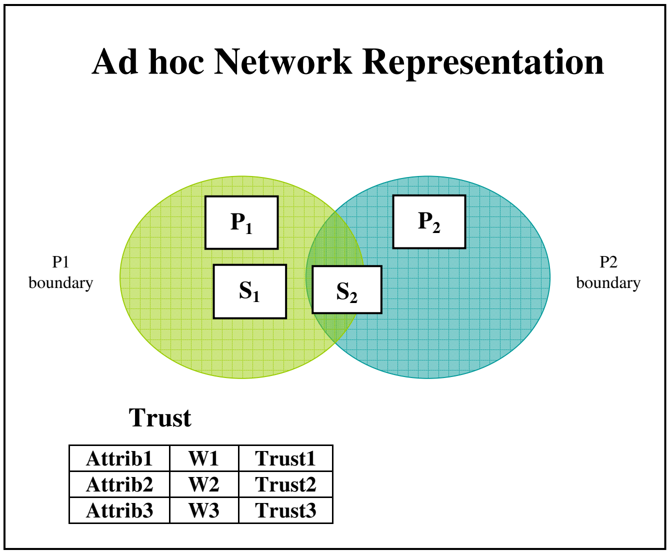


Figure 8. Variation of trust level as number of positive interactions increase

The behavior of current dynamic trust models and the trust level at any given time can be simplified, as shown in figure 8, as a function of the number of positive interactions. Although the concept of trust can be easily described as linear, in reality this is not the case. Different models exist and the majority are non-linear.

Rodriguez (2009) presented a method to enhance current trust data replication mechanisms [19]. Figure 9 depicts the key research scenario used for developing the replication approach. The ad-hoc network consists of two primary and two secondary devices. They are represented by P and S respectively. Each secondary device is at least directly accessible by one primary. In the case of S_2, P_1 and P_2 can directly access it. In this scenario, S_2 is attempting to communicate with S_1. S_2 initiates a connection at t_1, and S_1 simply responds. By using existing trust models, S_1 has the ability to determine the trust level of S_2 by querying another device, either primary or secondary. The resulting value comes from a trust table that was updated at some earlier time, t_0. The interval (t_0,t_1) represents the opportunity a malicious entity has to exploit any trust information exchange between devices comprising the network.



Attrib1	W1	Trust1
Attrib2	W2	Trust2
Attrib3	W3	Trust3

Figure 9. Ad hoc Network Representation

The Byzantine Generals' model, in particular the 3m+1 solution where m is the number of traitors, was used to develop the capability to reach an agreement by P and S as to what the true trust value of S_2 is, represented as $\prod_{\text{trust\_level}} (S_2)$. At t_0 and t_1, $\prod_{\text{trust\_level}} (S_2)$, as seen by P_1, is high. However, at t_1, $\prod_{\text{trust\_level}} (S_2)$, as seen by P_2 is low due to detected malicious behavior, so P_2 needs to update P_1 with this information and S_1 needs to obtain this update before deciding whether to allow the communication initiated by the now malicious entity S_2. If this update is not successfully propagated, S_1 and the rest of the network could be compromised.

The work performed through simulations resulted in a unique approach that will be used as the core for future enhancements. This approach relies on a trust topology. The trust level of each device is captured and added to an internal table, defined as T_{Dn} where D Є{P,S} and $1< n <\infty$. Given $\prod_{\text{trust\_level}}$ (Dn), an overall system trust topology, defined as T, can be represented as follows: $\prod_{\text{trust\_level}}(T_{D1}) \cap \prod_{\text{trust\_level}}(T_{D2}) \cap \ldots \cap \prod_{\text{trust\_level}}(T_{Dn})$ where D Є{P,S} and $1< n <\infty$. This enables the detection of potential malicious entities regardless of whether the compromised device's trust table is altered.

In the above key scenario at t_0, $T_{P1} = T_{P2} = T_{S1} = T_{S2} = \{P_{1H}, P_{2H}, S_{1H}, S_{2H}\}$. At t_1, S_2 is compromised and $T_{S2} = \{P_{1H}, P_{2H}, S_{1H}, S_{2L}\}$. At t_2, $T_{P2} = Ts_2$ while the trust table for S_2 is being changed to reflect a compromised trust table $T_{S2C} = \{P_{1H}, P_{2H}, S_{1H}, S_{2H}\}$. Once S_{2C} attempts to connect to S_1, S_1 sends a trust validation request to P_1, which in turn asks P_2 to further validate the trust table. At this point, a conflict arises when S_{2L} Є T_{P2} and S_{2H} Є T_{P1}. The lower level S_{2L} wins and triggers a full update of the trust topology with a new value of S_{2B}. S_2 is now isolated and no communication is allowed to or from it.

<u>High level Replication Algorithm</u>

```
[$S_2$ attempts to communicate with $S_1$]
[If  $Ts_1$_Elapsed_Time_Since_Last_Update >  Update_Threshold]
    *[$S_1$ requests $Tp_1$]*
    *[$S_1$ requests $Tp_2$]*
    [If  $Tp_1$= =$Tp_2$]
        [If   $Tp_{1(S2)}$ != $S_{2C}$]
            [Allow communication between $S_2$ and $S_1$]
        [Else]
            [Update systems topology]
            [Do not allow communication between $S_2$ and $S_1$]
        [End If]
    [End If]
[Else]
        [$P_1$ sends $Tp_1$ to $S_1$]
[End If]
```

[Dn requests T_{Di}]

```
[If $T_{Dn}$_Elapsed_Time_Since_Last_Update > Update_Threshold]
            [Dn requests $T_{Di}$]
```

Figure 10. Trust Replication Algorithm

Although the results above are promising, it is obvious that a potential denial of service attack can be launched by simply marking a node in the trust table as compromised and sending the update to all nodes. This is due to the fact that the algorithm assumes the lowest level of trust as the real trust level of the node. To counteract this, an additional step was added. This step relies on full mesh verification, when possible, where S_1 can get the trust table from both P_1 and P_2. At this point, the 3m+1 solution to the Byzantine Generals' Problem fully resolves any conflict by utilizing the majority value. In other words, S_1 is now receiving three different trust tables, two of them showing S_2 as compromised. In a scenario where S_2 can be trusted and another node inserts a trust table showing S_2 as compromised, the majority value approach will determine S_2 can be trusted and the algorithm will have the ability to identify the source of the attack.

3 HTM Based Dynamic Trust Replication

The goal of the proposed research is to develop a self-managed dynamic trust replication mechanism for cognitive networks by leveraging advances in the area of bio-insipired artificial intelligence, particularly HTM. It will ensure enhanced secure behaviors in devices belonging to cognitive networks by emulating human-like trust-based interaction between them.

The lack of secure behaviors in cognitive networks is a direct effect of a higher level cause: the lack of robust security. For example, the handoff process involves sensing the medium looking for vacant channels and choosing the best one according to various criteria, thus incurring high latencies until the transmission is resumed. A malicious user trying to disrupt a TCP connection of a secondary user can perform a Primary User Emulation Attack (PUEA) to force a handoff in the Cognitive Radio Network (CRN). As the transport layer is not aware of the disconnection, it keeps sending data segments which are queued at lower layers but not transmitted and thus TCP segments can be delayed or even lost. As the TCP sender is allowed to transmit new data upon reception of acknowledgments, loss or delay of segments can lead to a period of inactivity of the former.

It is well known that TCP triggers a retransmission timer (RTO) for each outstanding segment, which determines the time the sender waits for the corresponding acknowledgment before considering the segment has been lost. If the retransmission timer expires for a given segment, the TCP sender retransmits it and reduces the congestion window, since it is considered as a signal of congestion. The value assigned to the retransmission timer depends on the estimation of the round trip time (RTT) performed by the TCP sender and therefore, if the handoff period is large enough, it will lead to the expiration of many timers and the degradation of the throughput.

When considering the autonomous nature of cognitive networks as an effect, its causes are:

- Distributed Functionality
- Heterogeneous Purpose

These two causes are interrelated. The limited resources and functionality of a particular device (i.e. heterogeneous devices) give rise to the need of sharing information and distributing tasks (i.e. distributed functionality). Of course these two conditions are desired conditions for the successful development of cognitive networks.

Cognitive networks are vulnerable to the same type of network attack methodologies used in wired, ad hoc wireless and traditional wireless networks. Of course, the additional layer of mobility and autonomy found in cognitive networks make these challenges more complex while increasing the likelihood of new type of attacks. As established by Mathur and Subbalakshmi (2007) in the book by Qusay H. Mahmoud titled Cognitive Networks: towards self aware networks, reliable and robust security models need to be developed [14]. The inherent decentralized nature of ad hoc networks and their collaborative services give rise to a series of vulnerabilities that can be exploited by malicious entities as described by Zia, and Zomaya (2006).

As networks of hosts continue to grow in size and complexity, existing vulnerability evaluation approaches are no longer adequate due to their lack of accuracy and time intensiveness. Zhang (2005), and Bhattacharga (2008) discuss how its size and complexity require a different approach since traditional scanning and manual attack graph generation are time consuming and prone to errors. Existing literature shows a number of attack graph generation approaches that are both automated and scalable. However, further research is required to increase their accuracy and decrease their cycle time.

Cognitive networks are susceptible to attacks at the link layer, specifically by malicious entities acting as primary users (Mahmoud, 2007). According to Mahmoud, primary user identification is very important for both centralized and decentralized cognitive networks and further research in the use of cryptographic primitives needs to be performed. Mathr and Subbalakshmi (2007) recently proposed a digital signature approach for centralized networks that could be further evaluated for use on decentralized cognitive networks.

Existing intrusion detection approaches do not adequately address the needs of cognitive networks. According to Herve (2007), since cognitive networks rely on knowledge acquisition and exchange between nodes, it is likely that the knowledge management processes will be the target of attacks. Dain (2003), and Ning (2003) describe the challenge of processing data from heterogeneous sources to build an accurate picture. Debar (2005) proposes a model where cognitive networks processes can be leveraged to provide additional threat response mechanisms.

3.1 Methodology

Interdisciplinary targeted research in the areas of cognitive ad hoc networks, bio-inspired artificial intelligence, network security, and others will be performed to further evaluate existing approaches to the problem and to identify a clear course of action for developing a solution that will significantly enhance the current state of the art in the area of self-managed secure behaviors in cognitive networks. A detailed gap analysis between existing documented partial solutions and the desired solution will be performed. The data will be used to streamline the research effort by focusing on key needs.

Research Area	Purpose
Cognitive Networks	Identify and evaluate the most recent advances
Bio-inspired – Collective Systems	Interactions between devices in Cognitive Networks
Bio-inspired - Behavioral Systems	Actions taken by individual components in Cognitive Networks
Bio-inspired – Immune Systems	Malicious Detection mechanism
Swarm Intelligence	Collective Systems Behavioral Systems Immune Systems

Figure11. Research areas and their purpose at a glance

Figure 11 provides a summary of the research areas and their overall purpose in it study. It is important to note the emphasis of swarm intelligence as the combination of collective systems, behavioral systems, and immune systems. They are the key subareas of bio-inspired artificial intelligence that show significant promise in accomplishing the goal of the study.

3.2 Key Research Scenario

Figure 12 provides a high level visualization of the key research scenario that will be utilized as the foundation for the study. Cognitive networks are comprised of heterogeneous decentralized devices that are constantly on the move. Their functions vary according to their purpose (e.g. GPS, sensors, communication, etc.).

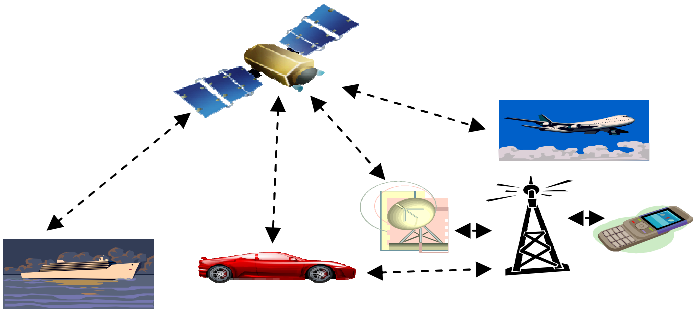


Figure 12. High-level depiction of Key Research Scenario for the study

Figure 13 describes some of the features for each type of device that will be utilized in the study. They include:

- GPS Data – spatial temporal data utilized to track location over time of devices in a cognitive network
- Radio – communication capability for interconnecting with other devices
- Mobile – ability to change location over time
- Technical Performance Parameters (TPMs) – Devices have constraints in the form of limited internal resources (e.g. memory, cpu, etc.). These are critical in monitoring end changing behaviors.

Network Entity	GPS Data	Radio	Mobile	TPMs
Cruise Liner	X	X	X	X
Satellite	X	X	X	X
Airplane	X	X	X	X
Cell phone	X	X	X	X
Tower		X		
Transmission Dish		X		

Figure 13. Cognitive network devices and their features

In this scenario, a malicious entity could gain access to any of the resources and launch a multitude of attacks. The important challenge will be to create a mechanism that will ensure secure self-managed behaviors that will prevent a full system compromise.

Research Area	Purpose	Goal
Cognitive Networks	• Cross-layer communication • Knowledge Management • Adaptive routing protocols • Documented security issues	1. Comprehensive understanding of intercommunication mechanisms 2. Comprehensive understanding of existing knowledge mechanisms that could be leveraged for the development of swarm intelligence 3. Comprehensive understanding of the conditions that could give rise to potential vulnerabilities
Bio-inspired – Collective Systems	Collective behaviors in biological organisms	Target insect colonies as the important framework for learning collective behaviors
Bio-inspired - Behavioral Systems	Individual behaviors in biological organisms	Correlation between individual behaviors and collective behaviors
Bio-inspired – Immune Systems	Biological organisms defense mechanisms against pathogen.	Framework for understanding individual behaviors in the presence of malicious entities
Swarm Intelligence	Collective Systems Behavioral Systems Immune Systems	The innovation will result from the intersection of the areas listed above.

Figure 14. Goals description according to the research area and their purpose.

Figure 14 provides a summary of the goals according the research areas and their purpose. This study will result in a significant enhancement to the state of the art in the area of self-managed secure behaviors in cognitive networks by improving autonomous dynamic trust data replication between devices.

4 Concluding Remarks

Dynamic trust replication remains a challenging problem that, if properly addressed, will significantly enhance the security and reliability of mobile Ad-hoc networks, in particular cognitive networks. HTM has shown promise in addressing some of the flaws commonly found in other bio-inspired artificial intelligence approaches, particularly the lack of temporal spatial awareness. Its use, combined with advances in other bio-inspired methodologies, has the potential to radically improve the current state of the art in self-managed secure behaviors in cognitive networks.

References:

[1] Chang, J., Wang, H. (2006). A dynamic trust metric for P2P systems. International Conference on Grid and Cooperative Computing Workshops. Los Alamitos: IEEE Computer Society.

[2] D' Arienzo, M., & Ventre, G. (2005). Flexible node design and implementation for self aware networks. International Workshop on Database and Expert System Applications (pp. 150-154). Los Alamitos: IEEE Computer Society.

[3] D' Arienzo, M., & Ventre, G. (2005). Flexible node design and implementation for self aware networks. International Workshop on Database and Expert System Applications (pp. 150-154). Los Alamitos: IEEE Computer Society.

[4] Gintis, H. (2000) Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Behaviors, Princeton University Press.

[5] Godefroid, P. 2007. Random testing for security: blackbox vs. whitebox fuzzing. In *Proceedings of the 2nd international Workshop on Random Testing: Co-Located with the 22nd IEEE/ACM international Conference on Automated Software Engineering (ASE 2007)* (Atlanta, Georgia, November 06 - 06, 2007). RT '07. ACM, New York, NY, 1-1.

[6] Hawkins, J, and Blakeslee, S.(2004). On Intelligence. New York: Holt Paperbacks.

[7] He, K., Feng, Z., & Li, X. An attack scenario based approach for software security testing at design stage. International Symposium on Computer Science and Computational Technology (vol. 1, no. 1, pp. 782-787). Los Alamitos: IEEE Computer Society.

[8] He, W., Xia, C., Wang, H., Zhang, C., & Ji, Y. A game theoretical attack-defense model oriented to network security risk assessment. International Conference on

Computer Science and Software Engineering (vol. 6, no. 6, pp. 498-504). Los Alamitos: IEEE Computer Society.

[9] He, W., Xia, C., Wang, H., Zhang, C., & Ji, Y. (2008). A network security risk assessment framework based on game theory. Second International Conference on Future Generation Communication and Networking (vol. 2, no. 2, pp. 249-253). Los Alamitos: IEEE Computer Society.

[10] Hedberg, S. (2007). Bridging the gap between neuroscience and AI. In the News, IEEE Intelligent Systems, vol. 22, no. 3, pp. 4-7, May/June, 2007.

[11] Hoey, J. (2001). Hierarchical unsupervised learning of facial expression categories. IEEE Workshop on Detection and Recognition of Events in Video (pp. 99). Los Alamitos: IEEE.

[12] Huaizhi, L., & Mukesh, S. (2006). A secure routing protocol for wireless ad hoc networks. Hawaii International Conference on System Sciences (pp. 225-234). Los Alamitos: IEEE Computer Society.

[13] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. ACM Transaction on Programming Languages and Systems (TOPLAS) (pp. 382-401). New York: ACM.

[14] Mahmoud, Q. (2007). Cognitive Networks: Towards Self-Aware Networks. New Jersey: Wiley.

[15] Moore, R (2007). PRESENCE: a human-inspired architecture for speech-based human-machine interaction. IEEE Transactions on Computers (vol. 56, no. 9, pp. 1176-1188). Los Alamitos: IEEE Computer Society.

[16] Rajendran, T., & Sreenaath, K. (2008, January). Secure anonymous routing in ad hoc networks. Paper presented at Compute 2008, Bangalore, Karnataka.

[17] Rathinam, A., & Padmini, S. (2007). Security assessment of power systems using artificial neural networks - A comparison between Euclidean distance based learning and supervised learning algorithms. International Conference on Computational Intelligence and Multimedia Applications (vol. 1, no. 1, pp. 250-254). Los Alamitos: IEEE Computer Society.

[18] Ricardo J. Rodriguez, & James A. Cannady. 2010. Automated risk assessment: A hierarchical temporal memory approach. In *Proceedings of the 9th WSEAS international conference on Data networks, communications, computers* (DNCOCO'10), Nikos Mastorakis, and Valeri Mladenov (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 53-57.

[19] Ricardo J. Rodriguez. 2009. Byzantine generals' problem driven dynamic trust replication method for cognitive mobile ad hoc networks (MANETs). In *Proceedings of the 8th WSEAS international conference on Data networks, communications, computers* (DNCOCO'09), Manoj Jha, Charles Long, Nikos Mastorakis, and Cornelia Aida Bulucea (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 31-34.

[20] Tikvati, A., Ben-Ari, M., & Ben-David Kolikant, Y. (2004). Virtual trees for the Byzantine generals algorithm. Proceeding of the 35th SIGCSE Technical Symposium on Computer Science Education (pp. 392-296). New York: ACM.

[21] Walters, R., Henderson, P., & Crouch, S. (2007). Selecting a distributed agreement algorithm. Proceedings o the 2007 ACM Symposium on Applied Computing (pp. 586-587). New York: ACM.

[22]Younan, Y., Piessens F., & Jousen, W. (2009). Protecting global and static variables from buffer overflow. International conference on availability, reliability, and security (pp. 798-803) Los Alamitos: IEEE Computer Society.

[23] Zhou, J., & Vigna, G. (2004). Detecting attacks that exploit application-logic errors through application-level auditing. Annual Computer Security Applications Conference (pp.168-178). Los Alamitos: IEEE Computer Society.

[24] Zia, T., & Zomaya, A. (2006). Security issues in wireless sensor networks. International Conference on Systems and Network Communication (pp. 40-43). Los Alamitos: IEEE Computer Society.

[25] Thompson, H., & Chase, S. (2005). The software vulnerability guide. Hingham, MA: Charles Rivera Media.

[26] Dowd, M., McDonald, John., & Schuh, Justin. (2006). The art of software security assessment: indentifying and preventing software vulnerabilities. Reading, MA: Addison-Wesley Professional.

[27] Numenta. (Producer/Director). (2008). Numenta HTM Workshop [Recording]. California: Numenta.