Danger Is Ubiquitous: Detecting Malicious Activities in Sensor Networks Using the Dendritic Cell Algorithm

Jungwon Kim, Peter Bentley, Christian Wallenta,
Mohamed Ahmed, and Stephen Hailes

Department of Computer Science, University College London,
Malet Place, London, U.K., WC1E 6BT
{J.Kim, P.Bentley, C.Wallenta, M.Ahmed, S.Hailes}@cs.ucl.ac.uk

Abstract. There is a list of unique immune features that are currently absent from the existing artificial immune systems and other intelligent paradigms. We argue that some of AIS features can be inherent in an application itself, and thus this type of application would be a more appropriate substrate in which to develop and integrate the benefits brought by AIS. We claim here that sensor networks are such an application area, in which the ideas from AIS can be readily applied. The objective of this paper is to illustrate how closely a Danger Theory based AIS - in particular the Dendritic Cell Algorithm matches the structure and functional requirements of sensor networks. This paper also introduces a new sensor network attack called an *Interest Cache Poisoning Attack* and discusses how the DCA can be applied to detect this attack.

Keywords: Danger Theory, Artificial Immune Systems, Sensor Networks, Interest Cache Poisoning Attack.

1 Introduction

Danger threatens living organisms every day of their lives. Intuitively, one might therefore suppose that a successful strategy in our immune systems would be to detect danger instead of relying solely on the detection of antigens that identify specific pathogens. A hotly debated hypothesis in immunology known as the Danger Theory [13] proposes just this. This theory suggests that the human immune system can detect danger in addition to antigens in order to trigger appropriate immune responses. The Danger Theory states that appropriate immune responses produced by the immune system emerge from the balance between the concentration of danger and safe signals within the tissue of a body, not by discrimination of self from non-self.

Danger also threatens modern computer networks every day. Aickelin *et al.* [1] presented the first in-depth discussion on the application of Danger Theory to intrusion detection and the possibility of combining research from wet and computer laboratory results. Their work aimed to build a computational model of Danger Theory in order to define, explore, and find danger signals. Greensmith *et al* [5] employed Dendritic Cells (DCs) within a Danger Theory based artificial immune system (AIS). DCs are a class of antigen presenting cells that ingest antigens or

H. Bersini and J. Carneiro (Eds.): ICARIS 2006, LNCS 4163, pp. 390 – 403, 2006.

protein fragments in the tissue. DCs are also receptive to danger signals in the environment that may be associated with antigens. Greensmith *et al* abstracted several properties of DCs that would be useful for anomaly detection and proposed the DC algorithm (DCA) to accommodate these properties. Recent work by the same authors [6] has also shown some initial results of using the DCA to detect port scanning. The outcome demonstrated the capability of the DCA as an anomaly detector.

As Hart and Timmis stated in [8], after a decade of research in the area of AIS, the researchers in the AIS community pose a question on whether there is a distinctive niche application area that AIS can provide unique benefits that is not presented by other existing approaches. They also highlighted a list of unique immune features that are currently absent from the existing AIS and other intelligent paradigms. We argue that some of these features can be inherent in an application itself, and thus this type of application would be a more appropriate substrate in which to develop and integrate the benefits brought by AIS. We claim here that sensor networks are such an application area, in which the ideas from AIS can be readily applied. The objective of this paper is to illustrate how closely Danger theory based AIS, in particular the DCA matches the structure and functional requirements of sensor networks.

The paper first reviews literature related to the Danger Theory based AIS. Section 3 illustrates how properties and functional requirements of sensor networks conform to an artificial tissue. Section 4 introduces a new sensor network attack called the '*Interest cache poisoning attack'* and section 5 discusses how the DCA can be applied to detect this attack. Finally, section 6 concludes this work with future work.

2 Danger Theory Based AIS

2.1 Previous Work

Since the first in-depth discussion of Danger Theory on the possibility of computing research [1], Bentley *et al* [3] introduced the concept of artificial tissue in order to adapt danger and safe signals (apoptosis and necrosis) thereby triggering artificial immune responses within an AIS. The authors stressed that the tissue is an integral part of immune function, with danger signals being released when tissue cells die under stressful conditions. Related work by Greensmith *et al* [5] employed DCs within AIS that coordinated T-cell immune responses. Kim *et al* [11] continued Greensmith *et al*'s work by discussing T-cell immunity and tolerance for computer worm detection. This work presented how three different processes within the function of T-cells, namely T-cell maturation, differentiation and proliferation could be embedded within the Danger Theory-based AIS. Twycross and Aickelin [15] provided a review of biological principles and properties of innate immunity, and showed how these could be incorporated into artificial models. In this work, authors addressed six properties of the innate immune system that would influence the capability of AIS. The same authors implemented the `libtissue` software that provides an innate immunity framework [16]. Finally, Le Boudec and Sarafijanovic [14] were also influenced by the idea of the Danger Theory, and chose to regard a packet loss in the network as a danger signal. Danger signals were used as co-stimulation signals confirming successful detection.

2.2 Dendritic Cell Algorithm

This paper focuses specifically on the Dendritic Cell Algorithm [5,6,7] of Greensmith *et al*, which abstracted a number of properties of DCs that are possibly advantageous to design AIS for anomaly detection.

In the human immune system, during the antigen ingestion process, immature DCs experience different types of signals that indicate the context (either safe or dangerous) of an environment where the digested antigens exist. The different types of signals lead DCs to differentiate into two types: mature and semi-mature. Chemical messages known as cytokines produced by mature and semi-mature DCs are different and influence the differentiation of naïve T-cells into several distinctive paths such as helper T-cells or killer T-cells. In order to employ these properties of DCs, Greensmith *et al.* categorised DC input signals into four groups – *PAMPs* (signals known to be pathogenic), *Safe Signals* (signals known to be normal), *Danger Signals* (signals that may indicate changes in behaviour) and *Inflammatory Cytokines* (signals that amplify the effects of other signals). When each artificial DC experiences the combination of these four different signal groups released by the artificial tissue, it interprets the context of ingested antigens by using a signal processing function, which weights each type of input signal differently. The output of a signal processing function determines the differentiation status of DCs (either semi-mature or mature).

3 Artificial Immune Systems Applied to Sensor Networks

The parallels between intrusion detection and immunity have long been the source of inspiration for AIS researchers, but conventional computer networks do not closely resemble the dynamic, distributed and fluid nature of organisms and their immune systems well. There is, however, a type of network that does share many of these features: sensor networks. In the following sections, we introduce this type of network and outline one popular routing protocol, known as *Directed Diffusion* [9].

3.1 Sensor Network Overview

Sensor networks are an emerging technology and research area in the rapidly growing field of ubiquitous computing [4], aimed at providing distributed and massively parallel monitoring in heterogeneous physical environments. Sensors are typically low-cost, limited capacity, mass production units, consisting of no more than (i) a sensing unit, (ii) a processing unit, (iii) memory, (iv) a transceiver and (v) a power unit [2]. Their aim is two fold: (i) to faithfully execute their intended task, and (ii) to efficiently manage their limited resources, such as energy, so as to maximise their lifetime. The following features of sensor networks distinguish them from traditional computing environments [2, 4]:

P1: Constrained resources – limited in physical capacity, bandwidth, cost, etc.

P2: High-density – number and density of sensor nodes can be several orders of magnitude higher than the mobile nodes in an ad hoc mobile network.

P3: Fidelity though redundancy – due to their physical constraints, individual nodes are prone to failure through deliberate attack or normal malfunction. The redundancy of nodes is used to compensate for this.

P4: Flexibility – aimed at operating under diverse conditions with minimal structured support, for example deployment in remote areas.

P5: Dynamic network topology - the topology may change often.

P6: Frequently data centric - IP addresses are not used, all nodes perform data-centric routing.

P7: Self-organising – network connectivity is often ad-hoc and dynamically maintained.

P8: Distributed computation – each node carries out simple data processing locally and sends out the partially processed data to other nodes. The chain of partial processing by individual nodes provides an aggregated solution.

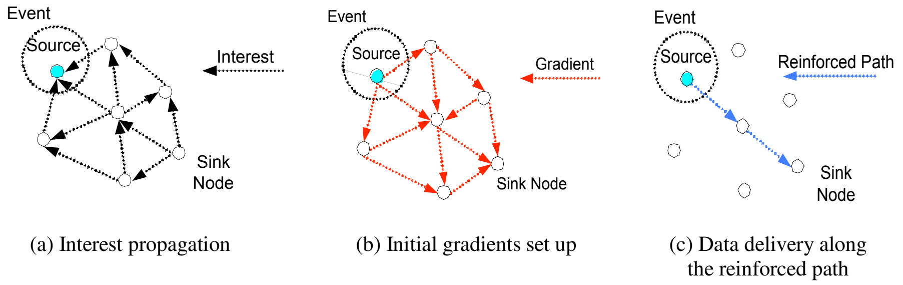
Together, these properties have provided the catalyst for a wide range of new applications, including environmental monitoring, disaster relief operations, military control/surveillance and health monitoring [2].

3.2 Directed Diffusion

In addition to the distributed and dynamic nature of sensor network hardware, one popular routing method is equally suggestive of natural immune metaphors: the *Directed Diffusion* protocol. This is a routing algorithm used to gather data sensed by a large number of sensor nodes and disseminate to a node that requests such data [9]. Directed Diffusion works in two phases, an initial exploratory phase that is followed by a reinforcement phase. Together these phases make up the three different stages discussed in Fig. 1.

The requesting node, referred to as the 'sink node' may request data from one or multiple other sensor nodes. As shown in Fig. 1(a), the sink periodically broadcasts its 'interest' packets (containing a description of the sensing task e.g. the regular reading of a patient's blood pressures) to its neighbours. Interest packets are then propagated throughout the whole network, resulting in creation of gradient fields representing the possible data flow paths from the source, back to the sink as shown in Fig. 1(b). Once the sink receives its requested data, it is then in a position to choose between its various neighbours by reinforcing the paths deemed most advantageous, for example based on the quality of service on the path that led to the neighbour, as shown in Fig. 1(c). As a result, though during the exploratory data packets are forwarded toward the sink node along multiple paths, the gradient refinement process chooses the most preferred path.

Reinforcements in Directed Diffusion come in two forms: positive and negative. Positive reinforcement encourages data flow along a given path, and the result is that data flows at a higher rate through the given path. In contrast, negative reinforcement discourages data flow along given paths, thereby reducing the rate at which data is sent through the path. The result is that the algorithms is dynamically able to tune its performance (with respect to the data flow path) based on arbitrary criteria.



(a) Interest propagation

(b) Initial gradients set up

(c) Data delivery along the reinforced path

Fig. 1. Directed Diffusion [9]

3.3 Wireless Sensor Tissue

Readers familiar with the field of AIS should find the properties of the sensor network using Directed Diffusion very familiar, because they mirror many of the properties of AIS algorithms. In this work we regard sensor networks as a suitable metaphor for the tissue of an organism - with diffusing packets acting as signals between cells. Using the work of Bentley [3] and Tycross [15] to aid this analogy:

- Tissue cells have limited processing, storage, and communication capacity; while a cell has its own capability of processing and storage, it takes a limited amount of input proteins such as cytokines or binds to a restricted number of neighbour cells. As described in *(P1)* sensor networks share these features.
- Biological tissue comprises a large number of cells. A tissue cell is the basic structural and functional unit, capable of functioning independently. A sensor network is similarly structured, see (*P2*).
- Each cell is prone to failure: cells in biological tissue are continuously exposed to pathogenic attacks, just as individual nodes of a sensor network are at risk, see *(P3)*. Later sections explain how an immune algorithm can integrate with a sensor network to help detect and overcome such attacks.
- The cells in living tissue move and reorganise themselves, just as nodes of a sensor network may move or be deployed in different places and have variable topologies, see *(P4)* and *(P5)*.
- Communication between biological cells is through the diffusion of signalling proteins and the matching of antigenic patterns; communication between sensor network nodes (using the Directed Diffusion protocol) is through diffusion and the matching of packets, see *(P6)*.
- Tissue cells are self-organising, growing without predetermined global control; the spatial and temporal information is passed by signals while receptors help the entire structure of the tissue develop. Likewise a sensor network automatically and dynamically forms its connectivity, see *(P7)*.
- Biological tissue cells are distributed, they work in parallel, signalling to each other to perform the desired functions. A sensor network is a truly distributed system with nodes that are processing in parallel and communicating with each other, see *(P8)*.

As discussed, the sensor network itself plays the role of artificial tissue and therefore the development of a separate artificial tissue as suggested in [3] and [15] is unnecessary.

4 Poisoning Sensor Networks

The analogy between sensor networks and tissue can also incorporate ideas of harm and damage. There are various types of vulnerabilities identified in sensor network environments that are often not found in conventional wired networks. This work focuses on vulnerabilities in sensor network routing protocols that rely on presence of limited capacity caches to keep a track of state of the network, for example the next hop for a packet**.** Directed Diffusion is one such protocol. Such protocols are typically optimised for nodes with limited resources and for specific applications, with little consideration for security.

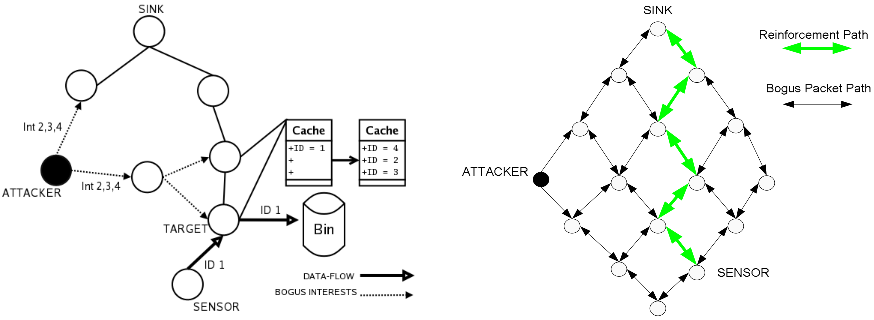
In their seminal work Karlof and Wagner [10] analysed diverse attacks against sensor network routing protocols and introduced some countermeasures. Notable attacks discussed include: Selective forwarding, Sinkhole attacks, Sybil attacks, Wormhole attacks, HELLO flood attacks and Acknowledge spoofing. In this paper, we introduce a new attack called the *'Interest Cache Poisoning Attack',* which can easily disrupt multiple data paths in a network. The attacks discussed in [10] exploit the vulnerabilities of sensor networks that are also found from mobile ad-hoc networks. In contrast, the interest cache poisoning attack reflects the vulnerability of data-centric approaches which are often adopted for routing in sensor networks.

Under the Directed Diffusion protocol, each node maintains an interest cache that records the history of received interest packets. Each entry contains an interest and gradient(s) towards neighbouring node(s) that have sent the interest packets, such that when a data packet arrives, a node looks up its interest cache in order to find the next hop for the data. If there is a matching interest, the node forwards the data packet to the neighbour node(s) indicated by the gradient(s). Otherwise the data packet is dropped. The basic idea of the interest cache poisoning attack is to inject fabricated interest packets to replace benign entries in the interest caches of other nodes. The attack is ideally aimed at nodes on established data paths that shall be referred to as the targets of the attack.

For example, in our study of Tiny Diffusion - an implementation of the Directed Diffusion protocol for real sensor nodes running the TinyOS[1], we found that: (i) An interest cache always has a fixed size and (ii) whenever a new interest packet arrives and the cache is full, the oldest entry is replaced. Therefore to realise a successful attack, the attacker can take advantage of the normal behaviour of the target by forcing it to drop the content of its cache. The attack works in two phases: First by flooding the target with bogus interests, thereby forcing it to drop those interests in its cache already. This leads to the second phase of the attack, when the requested data that was intended for distribution arrives, since the target no longer has gradients to those interested in it and will be forced to drop it.

[1] TinyOS is an open-source operating system designed for wireless embedded sensor networks. (http://www.tinyos.net/)

This process will result in the disruption of data packet delivery to the sink node. Ideally, a given cache entry needs to be wiped out before the first data packet from the source node arrives at the target node. Otherwise the attack may succeed but may not be able to completely suppress the data flow. Though mechanistically different, the effect of this attack is analogous to that of '*DNS cache-poisoning*' (http://en.wikipedia.org/wiki/DNS_cache_poisoning). However, we cannot use the same methods of protection against *DNS cache-poisoning* (i.e., randomised ports, restricted relaying, etc.) since these are aimed at the control plane and the *Interest Cache Poisoning Attack* is performed on the data plane.



(a) Interest Cache Poisoning Attack Overview

(b) Bogus interest packet propagation

Fig. 2. Interest Cache Poisoning Attack

Fig. 2 (a) shows the impact of the attack. The attacker sends out the bogus packets and fills up the cache of the nodes on the data path. The bogus interests will replace the original interest with ID 1. When the requested data with ID 1 arrives later, the target node will just drop it. This is because there is no matching entry in the cache. As shown in Fig. 2 (b), the attack will even be successful if the attacker is not next to the target node. The attack exploits the flooding behaviour of Directed Diffusion. Whenever a node receives a new interest packet it will rebroadcast it to all its neighbours. Hence, the bogus interest packets are spread and affect the caches of many nodes, eventually the cache of a target node. As a result, the impact of bogus packets can propagate over an entire network and disrupt multiple paths of data packet delivery.

5 Using the DC Algorithm to Detect Interest Cache Poisoning

Sensor networks using Directed Diffusion share a surprising number of similarities with biological tissue, including susceptibility to poison. Here we propose a security solution for sensor networks utilising Directed Diffusion with the aim of detecting cache poisoning attacks. The mechanism incorporates an immune algorithm inspired by the responsiveness of DCs in the human immune system to danger signals.

5.1 System Overview

Figure 3 shows the overall architecture of the Danger Theory based AIS, which employs the DC algorithm (DCA). Our Danger Theory based AIS comprises of two stages: (i) Detecting misbehaving nodes and (ii) detecting antigens and responding to the detected antigens. The DCA performs the first stage of the job, detecting misbehaving nodes. The second stage of the job involves sending immune cells and signals between the nodes of the sensor network. This may be performed by a different immune inspired algorithm such as the one introduced in [11]. This paper focuses on the first stage.

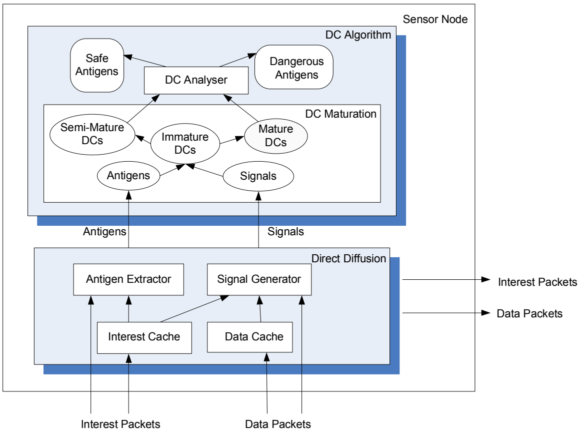


Fig. 3. DC algorithm and Directed Diffusion execute on a sensor node

A sensor node employing Directed Diffusion maintains two tables; the interest cache and the data cache and handles two types of packets; interest packets and data packets. While there are four possible sources of antigens and signals for input to the DCA, namely: (i) The interest cache, (ii) the data cache, (iii) interest packets and (iv) data packets. The signal generator and an antigen extractor are implemented as a sub-module of Directed Diffusion, thereby integrating the AIS into the protocol. When a packet arrives at a node, Directed Diffusion updates the interest and/or a data cache according to its local cache update rules [9], and extracts the signals and antigens from the packet(s) and/or cache(s). These are then passed to the DCA.

The immature DCs of the DCA sample the antigens and store them in their internal storage. They also combine various input signals using the signal weighting function shown in equation (1). The evaluation of the input signals results in output cytokines that differentiate between the immature DCs, to either become semi-mature or mature DCs. Antigens contained in semi-mature DCs are regarded as being collected under a normal condition, in contrast to the antigens stored by mature DCs that are collected under attack conditions. The DC analyser of the DCA reviews all the antigens stored in semi-mature and mature DCs and determines the state of each antigen as either “benign” or “malicious”.

5.2 Signals

The DCA uses the four different types of input signals discussed in Section 5.1. In the following, we introduce various input signals that can be collected from a sensor network environment in order to detect an interest cache poisoning attack. Signals are categorised into the four groups: (i) Danger Signals (*DS*), (ii) Safe Signals (*SS*), (iii) PAMP signals (*PS*) and (iv) Inflammatory Cytokines (*IC*). A detailed explanation on how these four categories are defined is presented in [5].

- *DS1 - Generated from the interest cache insertion rate*

This is the first Danger Signal collected from abnormal interest cache insertion rates. *DS1* signals are aimed at indicating that bogus interest packets have corrupted the interest cache of a node. In order to calculate this rate, a sliding time window is used to track the number of interest cache insertions per given time unit (such as 10 sec) and a total count is calculated by summing the window counts. After a minimum training period, the mean (μ) and standard deviation (σ) of the total count are calculated. *DS1* is generated with the concentration given by $(X_i - \mu) / \sigma$, where X_i is the count of in window *i*.

- *DS2 - Generated from the interest cache entry expiration*

There are two ways for an entry to be removed from the interest cache: (i) When its expiration time (a predefined time interval set by the sink node) has passed, or (ii) when the cache is already full and it is replaced by a new entry. Though a sink is able to overwrite its own entries in a cache by carelessly sending a large number of different interests during a short time interval, within in a well-behaved network, we do not expect this behaviour to be the norm. Therefore, the overwriting of entries long before their expiration time can indicate the presence of an attack. In order to identify such an event, the expiration field is checked whenever an entry is inserted. The concentration of a *DS2* signal is the time difference between the expiration time and the entry overwriting time. Overwriting a very recent entry will lead to a much stronger signal than overwriting a nearly expired entry.

- *SS - Generated from the arrival of data packets*

This measurement shows that the data requested by the sink node has been forwarded to a given node. The nature of the Safe Signal is to indicate normal data flow. The absence of a Safe Signal does not necessarily indicate the existence of an attack, but a Safe Signal can be used to suppress a false detection alert. The entry of a data cache, which records the data packet forwarded, would serve this purpose. Whenever a data packet that matches an interest in the interest cache arrives, it will be forwarded and recorded in the data cache. Therefore, whenever a new entry is inserted into the data cache, an *SS* is generated and the concentration of the *SS* is 1.0.

- *PS - Generated from the data delivery failure at the sink node*

A PAMP signal is a strong indicator of a pathogenic presence. For an interest cache poisoning attack, the failure of data delivery to the sink node strongly indicates the possibility of an attack. Though delivery failures may result from many factors such as node failures on the established path or the absence of sensor nodes generating the requested data - the PAMP signal definitively establishes that what was expected did not happen and can be used to launch further investigation. This relative difference of confidence in abnormal behaviour makes the PAMP signal stronger than a Danger

Signal. For this purpose, the failure of requested data delivery would cause the sink node to generate a *PS* signal. Unlike other signals, that are just generated locally and not forwarded to other nodes, the *PS* is forwarded to other nodes. In order to transport the *PS* signal, a re-sent interest packet is used, with concentration of 1.0.

- *IC1 - Generated from the changes in gradient directions*

This process aims to detect the onset of an attack through analysing the change in the gradient directions. Relative change in the number of gradients per neighbour indicates the addition or removal of paths to a data source by that neighbour and consequently the number of paths that go through the given neighbour. The normal behaviour of Directed Diffusion is such that if the majority of the maintained gradients point to a given neighbour, a node would expect that neighbour to be closer to the sink node than the other entities in the cache. This is because the only process that should result in an increase in the frequency of gradients to a given neighbour is the consequence of reinforcements applied to paths through that neighbour. In our analogy, inflammatory cytokine (IC) amplifies the effects of the other three types of signals but it alone is not sufficient to cause the maturation of a DC. *IC1* signals are generated by identifying bursts in the frequency of gradients to given neighbours. The concentration of *IC1* signals represents the magnitude of the changes. Though *IC1* alone is not strong enough to indicate an attack, i.e. it could be the result of a normal topology change; it still indicates a disturbance that should be noted. It therefore represents an *IC* and not a *DS*.

- *IC2 - Generated from data without matching interest cache entry*

The reception of a data packet that cannot be matched to an interest in the cache can be used as an indicator of a problem. Though this does not necessarily indicate the presence of an attack, for example as the result of different interest expiration times, it still identifies anomalous situations. The concentration of *IC2* is 1.0.

5.3 Antigens

From the view point of Danger Theory, antigens together with signals trigger immune responses. Antigens can originate from pathogens, the self or foreign cells. Immune cells attempt to bind antigens presented by semi-mature or mature DCs. When the receptors of immune cells bind to antigens passed by mature DCs, the immune cells become activated and later respond to new antigens binding to their receptors, i.e. killing antigens. In contrast, when the receptors of immune cells bind to antigens presented by semi-mature DCs, the immune cells become suppressed and later do not respond to new antigens binding to the receptors[2].

Likewise, the receptors of immune cells are used to find targets (antigens) of their immune responses. The AIS proposed in this work is required to have two types of responses. The first response is to identify an attacker node where a fabricated interest packet is created and sent out, and then to exclude this node from a sensor network. The second response is to identify bogus interest packets and then to stop forwarding them. For an interest cache poisoning attack, a node that is receiving bogus packets

[2] Or the receptors of immune cells binding antigens presented by semi-mature DCs will bind to the receptors of other immune cells and suppress the responses released by these other immune cells. Regulatory T cells are such immune cells.

(and thus its cache is being poisoned), might poison its neighbour nodes by forwarding the bogus packets. If the AIS excludes this kind of node from a sensor network, it runs the risk of disabling the entire network. In this case, a more desirable response could be to continue the delivery of genuine packets while stopping the forwarding of bogus interest packets. This work focuses on making the second response and hence regards interest packets as antigens. In future work we aim to add further antigens to trigger the first type of response – identifying an attacker node.

5.4 The Ubiquitous Dendritic Cell Algorithm

Detailed description of the original DCA is presented in [7] and a simplified pseudo-code of the ubiquitous DCA (UDCA) is shown in fig 4. UDCA is a variation of DCA that is designed to detect 'Interest Cache Poisoning Attacks' on sensor networks. UDCA has several properties that distinguish it from existing AIS. In the following section, we address the key elements of UDCA that could be particularly beneficial in detecting malicious activities in sensor networks, and their implementation in UDCA.

- UDCA attempts to collect signals from *multiple* data sources: Although multiple signals provide richer information to make a detection decision, they require temporal calibration. Line 8-14 of fig. 4 shows that a DC continuously calculates a new output cytokine with new signals and antigens collected at each DC maturing cycle (DC_Mat_Cycle). New output cytokines are then added to previously estimated ones until the CSM cytokine reaches a migration threshold. This allows a DC to collect signals indicating a possibly identical status of context despite being generated asynchronously. Hence, UDCA fine-tunes delays between multiple signals using a CSM value update with migration threshold.
- UDCA maps the context information delivered by signals with antigens in a *temporal* manner: antigens (interests) are gathered when signals are generated (see Signal_Generator and Antigen_Extractor at fig. 4). Depending on the type of signals, one or multiple antigens can be paired with a signal. For instance, in the UDCA (for SIG_new in Antigen_Extractor at fig.4), *DS2*, *SS* and *PS* will be paired with one interest packet triggering the signal generation. However, for *DS1*, *IC1* and *IC2,* all the interests that exist at an interest cache when these signals are generated will be selected as antigens. In this case, the antigen extractor collects antigens that are temporally close to signals since the signals are generated from the changes at multiple entries of interest caches or an absence of matching benign interest.
- UDCA combines multiple signals to judge an antigen context status: the diverse nature of signals contribute differently when judging an antigen context status. Empirical data obtained from immunologists' experimental results[3] suggest the weight values given in table 1. Equation (1) is a weighting function that determines the output cytokine by combining four types of input signals. This weighting function is used to handle a possible inconsistency existing between various signals. A given antigen can be judged by different signals in a

[3] These results were obtained by the research team led by Dr. Julie McLeod, Dr. Rachel Harry and Charlotte Williams at University of West England.

contradictory manner – "semi-mature" and "mature". In this case, the equation (1) determines a final decision by assigning a different weight to each signal. The line 10 – 19 of fig.4 shows this stage of UDCA processing.

```
PROCEDURE DC_Maturation(Ag_pop)
1  Let DC_Mat_Cycle = 1;
2  Creates a DC population, DC_pop;
3  A migration threshold value is randomly generated from a given range
4  Set a generated migration threshold value to each DC in DC_pop
5  Do
6 {
7    For each DC from DC_pop
8        Sample antigens, AGs, from Ag_pop, with replacement
9        Store sampled antigens to DC's internal antigen storage
10       Copy the signals paired with AGs to DC's internal signal storage
11       Calculate the concent. for CSM, MAT, SEMI-MAT cytokine of DC using (1)
12
13       Add CSM, MAT, SEMI-MAT cytok. to
14            total CSM, MAT, SEMI-MAT cytokine concent. respectively
15       If a total CSM cytokine concent. > an assigned migration threshold
16            If SEMI-MAT cytokine concent. > MAT cytokine concent.
17               DC is moved to semi-mature DC population, SEMI_MAT_DC_pop
18            else
19               DC is moved to mature DC population, MAT_DC_pop
20            endif
21            call DC_Analyser(SEMI_MAT_DC_pop, MAT_DC_pop)
22          endIf
23   endFor
24   Empty Ag_pop;
25   DC_Mat_Cycle++;
26 } while ( DC_Mat_Cycle < Max_DC_Cycle )

PROCEDURE DC_Analyser(SEMI_MAT_DC_pop, MAT_DC_pop)
1   For each antigen Ag from SEMI_MAT_DC_pop and MAT_DC_pop
2     Counts the number of times presented by SEMI_MAT_DC or MAT_DC
3     If SEMI_MAT_COUNT > MAT_COUNT
4     Ag is malicious
5     else
6     Ag is benign
7     endIF
8   endFor
9   For each DC from SEMI_MAT_DC_pop and MAT_DC_pop
10  Reset a migration threshold value of DC
11  Set CSM, MAT, SEMI_MAT cytokine concent. of DC to be 0
12  Set total CSM, MAT, SEMI_MAT cytokine concent. of DC to be 0
13  Empty antigen and signal storages of DC
14  Move the DC to DC_pop from  SEMI_MAT_DC_pop or MAT_DC_pop
15  EndFor

PROCEDURE Signal_Generator(Interest Cache, Data Cache, Packets)
1   Generates a new signal, SIG_new  // as described in section 5.2
2   If  SIG_new is generated
3       Call Antigen_Extractor(Interest Cache, SIG_new)
4   endIf

PROCEDURE Antigen_Extractor(Interest Cache, SIG_new)
1   Check through an Interest Cache
2   Select interests matching to SIG_new
3   Each selected interest becomes an antigen
4   Add pairs of an antigen with SIG_new to Ag_pop
```

Fig. 4. Pseudo code of the UDC algorithm to detect malicious activites

Table 1. Suggested weights used for Equation (1), which is a signal weighting function [6]. W_P, W_D, W_S, C_P, C_D, C_S are weights and concentrations of PS, DS, SS respectively.

Weight	csm	semi	mat
W_P	2	0	2
W_D	1	0	1
W_S	2	3	-3

$$C_{[csm,semi,mat]} = \frac{(W_p{*}C_p)+(W_s{*}C_s)+(W_D{*}C_D)}{|W_p|+|W_s|+|W_D|} * \frac{1+IC}{2} \qquad (1)$$

- UDCA employs a population of DCs to determine the final antigen context status: as shown DC_Analyser procedure of UDCA in fig. 4, the context status of each antigen is determined by the collective decisions of multiple DCs'. Each DC samples antigens and its migration threshold values are set differently (see line 2-3 of fig.4). These allow each DC to judge the context of one antigen differently and the final decision on a given antigen is therefore made from the aggregations from multiple DCs.
- UDCA does not employ a pattern matching based detection: UDCA concentrates on identifying bogus interest packets and filtering them out. This is another different trait from other existing AISs, which usually employ pattern matching to detect an on-going attack. UDCA detects an attack by examining how much a given node is misbehaving via generated signals. It then collects data (=antigens) for the next AIS algorithm to perform a pattern matching detection, which is required to produce responses. In responding, an AIS needs to react to a malicious antigen before it damages a monitored system and causes generations of signals. It is necessary for an artificial immune responder to have a pattern matching based detection. Therefore, UDCA plays the role of the innate immune system that presents the context information with matching antigens to the adaptive immune system [3], [15].

6 Conclusion

This work introduces the concept of sensor networks as a new application area for AIS research and argues that some AIS features are inherent in sensor networks. We illustrate how closely a Danger Theory based AIS, in particular the dendritic cell algorithm (DCA), matches the structure and functional requirements of sensor networks. This work also introduces a new sensor network attack called an interest cache poisoning attack and discusses how the DCA can be applied to detect an interest cache poisoning attack.

Currently we have implemented a number of different versions of an interest cache poisoning attacks by varying the bogus packet sending rates, the number of sink node interest subscriptions and the location of an attacker. In addition, various types of signals introduced in this paper have been being generated. The attacks and the signal generator have been being implemented under a network simulator, J-Sim (www.j-sim.org) and TOSSIM (www.cs.berkeley.edu/~pal/research/tossim.html). As discussed in this paper, UDCA appears to be an attractive solution to filter out bogus packets but the more detailed features of UDCA need to be further investigated. In future work, we aim to thoroughly study the appropriateness of a weight function used, the sensitivity analysis of various parameters, and the efficiency required to be used in a limited environment like a sensor node.

Acknowledgments

This project is supported by the EPSRC (GR/S47809/01).

References

1. Aickelin, U. Bentley, P., Cayzer, S., Kim, J., and McLeod, J.: Danger Theory: the Link between AIS and IDS. In Proc. of the 2nd Int. Conf. on AIS (ICARIS-03), (2003) 147-155.
2. Akyildiz, I. F. et al.: A Survey on Sensor Networks, IEEE Communication Magazine, Aug., (2002) 102-114.
3. Bentley, P., Greensmith, J., and Ujjin, S. : Two ways to grow tissue for AIS, In Proc. of the 3rd Int. Conf. on AIS (ICARIS-05), Springer-Verlag (2005) 139-152
4. Estrin, D., Cullar, D., Pister, K., and Sukhatme, G. : Connecting the Physical World with Pervasive Networks, Pervasive Computing, (2002) 59-69
5. Greensmith, J., Aickelin, U. and Cayzer, S. : Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Intrusion Detection, In Proc. of ICARIS-05, Springer-Verlag (2005) 153-167
6. Greensmith, J., Twycross, J., and Aickelin, U.: Dendritic Cells for Anomaly Detection, In Proc. of IEEE Cong. on Evolutionary Computation (CEC-06), Vancouver, Canada (2006).
7. Greensmith, J. et al.: Articulation and Clarification of the Dendritic Cell Algorithm, submitted to ICARIS-2006
8. Hart, E., and Timmis, J. : Application Areas of AIS : The Past, The Present and The Future, In Proc. of the 3rd Int. Conf. on AIS (ICARIS-05), (2005) 483-497
9. Intanagonwiwat, C. et al : Directed Diffusion for Wireless Sensor Networking, IEEE/ACM Trans. on Networking, Vol.11, No.1, Feb (2003) 2-16.
10. Karlof, C., and Wagner, D. : Secure routing in wireless sensor networks: attacks and countermeasures, Ad Hoc Networks, (2004) 293-315
11. Kim, J., Wilson, W., Aickelin, U. and McLeod, J. : Cooperative Automated Worm Response and Detection ImmuNe Algorithm (CARDINAL) inspired by T-cell Immunity and Tolerance, In Proc. of the 3rd Int. Conf. on AIS (ICARIS-05), (2005) 168-181
12. Kim, J. et al, Immune System Approaches to Intrusion Detection – a Review, under review.
13. Matzinger, P.: Tolerance, danger and the extended family. Annual Reviews in Immunology, 12 (1994) 991-1045
14. Sarafijanovic, S. and Le Boudec, J. : An AIS for misbehaviour detection in mobile ad-hoc networks with virtual thymus, clustering, danger signals and memory detectors, In Proc. of the 2rd Int. Conf. on AIS (ICARIS-04), Springer-Verlag (2005) 342-356
15. Twycross, J. and Aickelin, U. : Towards a conceptual framework for innate immunity, In Proc. of the 3rd Int. Conf. on AIS (ICARIS-05), Springer-Verlag (2005) 153-167
16. Twycross, J. and Aickelin, U.: libtissue – implementing innate immunity, In Proc. of the CEC-06, Vancouver, Canada (2006) to appear.