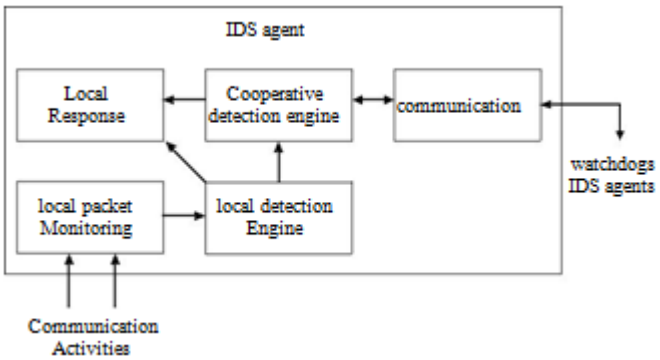

Figure 4: The building blocks of the IDS client existing in each sensor node.

- Network Monitoring: Every node performs packet monitoring in their immediate neighborhood collecting audit data.
- Decision Making: Using this audit data, every node decides on the intrusion threat level on a host-based basis. Then they publish their findings to their neighbors and make the final collective decision.
- Action: Every node has a response mechanism that allow it to respond to an intrusion situation.

Based on these functions we build the architecture of the IDS client based on five conceptual modules, as shown in Figure 4. Each module is responsible for a specific function which we describe in the sections below. The IDS clients are identical in each node and they can broadcast messages to clients in neighboring nodes to listen. The communication amongst the clients allows us to use a distributed algorithm for the final decision on the intrusion threat.

A. Local Packet Monitoring

This module gathers audit data to be provided to the local detection module. Audit data in a sensor network IDS system can be the communication activities within its radio range. This data can be collected by listening promiscuously to neighboring nodes' transmissions.

B. Local Detection Engine

This module collects the audit data and analyzes it according to given rules. As we said in Section III-B, specification based detection is most appropriate for sensor networks, so the local detection engine stores and applies the defined specifications that describe what is a correct operation and monitors audit data with respect to these constraints.

C. Cooperative Detection Engine

If there is an evidence of intrusion, this module broadcasts the state information of the local detection process to the neighboring nodes. The same module in each node collects this information from all the neighboring nodes and applies a majority rule to conclude whether there is an intrusion or not. The input from the local detection engine is also counted in for this conclusion.

D. Local Response

Once the network is aware that an intrusion has taken place and have detected the compromised area, appropriate actions are taken by the local response module. The first action is to cut off the intruder as much as possible and isolate the compromised nodes. After that, proper operation of the network must resume. This may include changes in the routing paths, updates of the cryptographic material or restoring part of the system using redundant information distributed in other parts of the network. Autonomic behavior of sensor networks means that these functions must be performed without human intervention and within finite time.

Depending on the confidence and the type of the attack, we categorize the response to two types:

- **Direct response:** Excluding the suspect node from any paths and forcing regeneration of new cryptographic keys with the rest of the neighbors.
- **Indirect response:** Notifying the base station about the intruder or reducing the quality estimation for the link to that node, so that it will gradually lose its path reliability.

5. Experimental Evaluation

I have simulated a sensor network of 100 nodes placed uniformly random in order to test our proposed intrusion detection system. The network density was chosen so that each node has 8 neighbors on the average. Each time, we chose at random one link A → B and programmed node B to launch a selective forwarding attack, while node A was sending packets to it, at a given rate. This way we could have the watchdogs of that link A → B apply the intrusion detection and monitor the behavior of node B. With probability pD , node B was dropping the packets that were forwarded to it. Finally, we set the threshold value for the percentage of packets dropped over a period w to t = 20%. Above this threshold, each watchdog was generating an alert. Packets dropped at a lower rate were calculated to other factors, such as collisions or node failures, and did not produce an intrusion alert.

First we tested how the ratio of W and w effects the accuracy on intruder identification. The results are depicted in Figure 5, for 100 repetitions of the experiment. As we said in Section III-C, W must be bigger than w, so we did not simulate the case of W/w < 1. False negative rate represents the rate at which events are not flagged intrusive by the collector although the drop rate is higher than the threshold and the attack exists. If packets are dropped at a rate higher than the threshold t, then ideally, all windows W at the collector should give an alarm. However, since packets are dropped probabilistically, there might be the case that during a window w of some watchdogs, the dropped packets are less than t = 20%, and no alert is produced by those nodes. Then, the majority rule over a window W will not be satisfied, which will give no final alarm, producing a false negative.

This is less probable to happen as pD increases compared to t. In this case, the probability that during a window w the

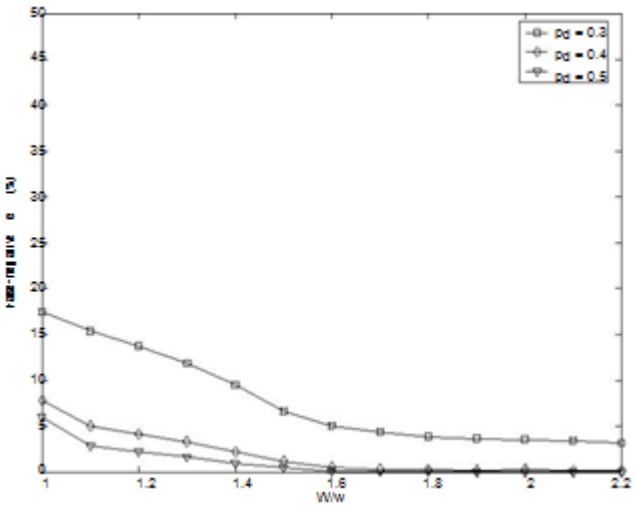

Figure 5: False-negative rate for different ratios of window length **W** to **w**

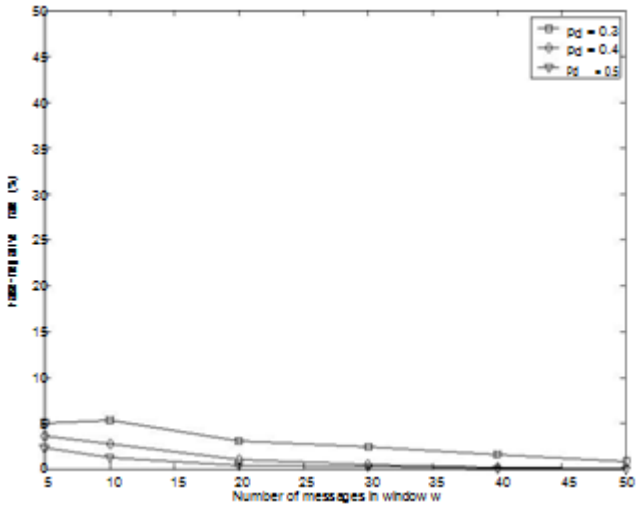

Figure 6: False-negative rate for different window lengths W.

dropped packets are less than t resulting in a false negative is lower, and hence the better accuracy in detecting the attack.

We see from Figure 5 that as the window length W increases, the false negative probability decreases. This is because the collector can have a more accurate estimation as it gives more time to the watchdogs to produce their alarms. However, we cannot take W to be a very large quantity, since that would delay the detection of a compromised node. Therefore, for the rest of the experiments we fixed W = 2w.

Next we tested how the window length w affects the accuracy on intruder identification. All watchdogs are required to have the same window length. Given a steady packet rate, we measure this length in number of packets. Figure 6 shows the false negative rate for different number of packets monitored by the watchdogs. Then, for a fixed simulation time, we measured the number of final intrusion alerts produced by the cooperative engine at the collector. For the given window W (= 2w), each watchdog gathers the alerts broadcasted by the rest of them and applies the majority rule to produce a final decision.

Figure 6 shows that the false negative rate is reduced as the window length w is increased. For bigger w, more packets are monitored, and therefore, each watchdog has a better estimation of the drop rate and alerts are more successfully produced resulting in a cooperative detection at the collector. In the rest of the cases, the drop rate over the time period w for a watchdog may be statistically below the threshold, and no alert is produced. If this is true for more than half of the watchdogs, the majority rule fails and no detection is made.

Figure 7 depicts the number of alerts from the collector as a function of the drop probability pD . Two thresholds of 20% and 10% have been assumed for the local detection at the watchdogs. In all experiments we took W = 2w. The simulation time is fixed for 1000 repetitions and we set w to be long enough for 30 messages to be monitored at each watchdog.

Maximum number of final alerts that could be produced by the collector is 16, since this is the maximum number of windows W that fit in the fixed simulation time. For drop probabilities below the threshold a small number of alerts is produced. This is the number of *false positives* and ideally it should be zero. Since the packets are dropped probabilistically, there are cases where more than 20% (or 10% respectively) of the packets are dropped, even if the drop probability is lower. However, on the average, the cooperative mechanism produces a small number of false positives and this effect is shown clearly on smaller drop probabilities. For example, if we set the threshold t = 20% and assume that packets are dropped at a lower rate $p_D = 0.1$, then the graph indicates that the false positives will be 0.52, which is a rate of $0.52 \times 100/16 = 3.25\%$.

6. Conclusions

In this paper we have introduced a model for distributed intrusion detection in sensor networks which is designed to work with only partial and localized information available at each node of the network. Nodes collaborate and exchange this information with their neighbors in order to make a correct decision on whether an attack has been launched. We focused on routing because it is the foundation of sensor networks. In particular, we demonstrated how our IDS system can be used to detect blackhole and selective forwarding attacks, producing very low false-negative and false-positive rates. We also provided a set of general principles that an IDS system for sensor networks.

References

[1] S. Camtepe and B. Yener, “Key distribution mechanisms fo r wireless sensor networks: a survey,” Rensselaer Polytechnic Instit ute, Troy, New York, Technical Report 05-07, March 2005.
[2] E. Shi and A. Perrig, “Designing secure sensor networks, ” IEEE Wireless Communications, vol. 11, no. 6, pp. 38–43, December 2004.
[3] L. Lazos and R. Poovendran, “Serloc: Robust localizatio n for wireless sensor networks,” ACM Transactions on Sensor Networks, vol. 1, no. 1, pp. 73–100, 2005.

[4] T. Dimitriou and I. Krontiris, Security in Sensor Networks. CRC Press, 2006, ch. Secure In-network Processing in Sensor Networks, pp. 275– 290.
[5] S. Ganeriwal, S. Capkun, C.-C. Han, and M. Srivastava, "S ecure time synchronization service for sensor networks," in Proceedings of the 4th ACM workshop on Wireless security (WiSe '05), 2005, pp. 97–106.
[6] [6] A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sens or networks," in Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks (Q2SWinet '05). ACM Press, October 2005, pp. 16–23.
[7] Onat and A. Miri, "An intrusion detection system for wi reless sensor networks," in Proceeding of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 3, Montreal, Canada, August 2005, pp. 253–259.
[8] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrus ion detection for routing attacks in sensor networks," International Journal of Distributed Sensor Networks, 2005.