Implementation and Evaluation of Distance-Based Message Authentication

Tony Chung and Utz Roedig
School of Computing and Communication, Infolab21, Lancaster University, UK
{a.chunglu.roedig}@lancaster.ac.uk

***Abstract*—A new generation of WSN communication transceivers are now available that support time-of-flight distance measurement. This measurement can be inseparably integrated with message transmission making it possible to authenticate messages based on distance. In this paper we present a practical implementation of Distance Based Message Authentication (DBMA) for WSNs using Nanotron NA5TR1 transceivers. We show that DBMA can be used to reject messages sent from outside a secure (trusted) area. With DBMA, messages can be authenticated without involving costly cryptographic algorithms. The DBMA implementation is evaluated in two different deployment scenarios. Distance measurement errors and their impact on the size of the required secure area are evaluated. Furthermore, we present methods to reduce the size of the required secure area.**

I. Introduction

A new generation of communication transceivers are now available that support time-of-flight distance measurement. As RF propagation speed is a constant, senders cannot decrease propagation delay to produce a shortened distance measurement. It is also possible to ensure that distance measurement and message transmission are performed inseparably. A receiver can thus reject a message if the measured distance to the sender is above a set threshold. We refer to this method of authentication as Distance Based Message Authentication (DBMA). DBMA may be used as an additional layer of defense to increase protection levels or as a replacement for authentication mechanisms based on traditional cryptography.

DBMA is only useful in scenarios where potential attackers can be excluded from the WSN deployment area. This is certainly not possible in all WSN deployments; however, many deployments exist where physical access is restricted and monitored. Examples are WSN deployments in buildings, factories or military installations. In these scenarios DBMA prevents injection of messages from potential attackers located outside of the WSN deployment area.

Physical access control requires the implementation of a security boundary, normally in form of a fence or a wall. The question is where the fence in relation to the WSN deployment has to be placed such that DBMA can be effective. At first glance it seems the fence must be placed such that the distance from each node to the fence is larger than the distance of each node to its communication peers in the WSN. Unfortunately, the measured distances in a real-world DBMA implementation are erroneous and may be larger (but never shorter!) than the Euclidean distance as signals may follow non-line-of-sight paths. The fence must therefore be placed using *measured* distances and not Euclidean distances. Thus, undesired additional space between deployment and fence is required to compensate for distance measurement errors.

Consider a network with two nodes, A and B, that have the Euclidean distance of $5m$. A secure area could be constructed using the union of two circles with radius $5m$ drawn around A and B. Nodes A and B will reject communication from further away than $5m$. Now consider that the *measured* distance between A and B is between $5m$ and $6m$. In this case A and B would reject genuine transmissions as they *appear* to originate from outside the secure area. By constructing the secure area using $6m$ circles around A and B the problem can be corrected. Note that it is not acceptable to simply subtract expected errors from measurements as an attacker may be able to produce error free measurements.

Obviously, distance measurement errors have an impact on fence positioning and the resulting size of the secure area. A secure area cannot be arbitrarily large in a practical WSN deployment; it may be too expensive or just impossible to acquire land. It is therefore the aim of this paper to analyze the required secure area size of a practical DBMA implementation and to investigate methods useful to reduce the secure area size. The specific contributions of this paper are:

- *Implementation:* We present an implementation of DBMA using Nanotron NA5TR1 [1] transceivers.
- *Evaluation:* We evaluate distance measurement errors and resulting secure area size in two different WSN deployments.
- *Optimisation:* We reduce the secure area size by pruning links with high distance measurement errors from the WSN topology.

The paper is organised as follows. Related work is

discussed next. Section III introduces our RTTMAP protocol used to implement DBMA. We detail our implementation of RTTMAP for the Nanotron NA5TR1 transceiver in Section IV. The impact on secure area size from distance measurement error is evaluated in Section V. Section VI evaluates methods to reduce the secure area size. We conclude the paper in Section VII.

II. Related Work

Traditional cryptographic authentication is difficult to implement on resource-constrained sensor nodes. Existing work aims to improve efficiency of cryptography by employing dedicated hardware or optimised algorithms (For example, [2], [3], [4], [5]). Our work provides an alternative authentication mechanism which does not (similar to [6]) expose cryptographic mechanisms to attackers. The problem of trustworthy distance measurements has been discussed in the context of positioning systems [7]. We now discuss these topics.

The IEEE 802.15.4 [2] standard defines Advanced Encryption Standard (AES) message encryption and authentication on the link-layer. The cryptographic algorithms are executed by specialized hardware within the transceiver chip. Hu et al. [3] propose the integration of a Trusted Platform Module (TPM) chip on a sensor node. This specialised hardware is able to execute cryptographic algorithms efficiently. Liu et al. [4] and Szczechowiak et al. [5] both investigate libraries for elliptic curve cryptography in wireless sensor networks. Elliptic curve cryptography is the most efficient way of providing public key cryptography. Furthermore, the libraries provide a number of optimization switches which can turn specific optimizations on or off. Thus, it is possible to implement efficiently strong cryptography on resource constrained sensor nodes.

Martinovic et al. [6] propose a method for message authentication which does - similar to DBMA - not rely on cryptographic mechanisms for message authentication. Sensor nodes are prevented from receiving unauthorised transmissions. Nodes in the network monitor transmissions and jam signals that are perceived to be transmitted by unauthorised devices. Battery-depletion attacks are prevented as unauthorised messages are not processed by nodes. Thus, the work by Martinovic et al. is close to the presented DBMA work as it achieves the same goal using a different mechanism.

Capkun et al. [7] discuss generally accuracy of positioning system if attackers are present. Internal attackers can report false position and distance information in order to lie about their position. External attackers can spoof the measured positions and distances. It is pointed out that out of all available positioning methods radio frequency (RF) based time of flight (ToF) measurements have the best security properties. RF signals travel at the speed of light and an attacker can only increase but not decrease the measured ToF between nodes. The DBMA implementation presented and evaluated in this paper makes use of RF ToF distance measurements.

In our previous work [8] we have described the principal mechanisms of DBMA. We also described a protocol called RTTMAP that can be used to implement DBMA. However, our previous work does not provide insight into real-world implementation of DBMA or an analysis of its performance in a deployment scenario.

III. Distance Based Message Authentication

In this section we motivate and explain the concept of DBMA and we describe briefly the protocol RTTMAP that can be used to implement DBMA.

A. DBMA Scenario

We are experimenting with a physical intrusion detection system used to secure buildings. Nodes equipped with sensors, such as infra-red detectors or door contacts, report observations to a sink for data analysis, logging and alarm generation. The network currently relies on cryptographic safeguards. Each node signs messages using a unique symmetric key shared with the sink.

This approach is feasible but has some drawbacks. Firstly, a key management protocol has to be executed in the network consuming scarce bandwidth and energy resources. Secondly, scarce energy and computation resources are used to cryptographically authenticate messages. Thus, an attacker may drain energy by injecting unauthorised messages which need to be cryptographically processed before they can be discarded.

DBMA can be used to augment the existing cryptographic mechanisms or it could be used to replace them entirely in some scenarios. DBMA aims to solve the two previously outlined problems. First, DBMA does not require costly key distribution. Second, DBMA prevents costly processing of unauthorised messages. The outlined application scenario can use DBMA as a secure area can be enforced. The WSN itself is used to restrict physical access to a secure area.

B. DBMA Implementation Considerations

DBMA uses the distance between two nodes as a security parameter. If the RF communication transceiver is used for DBMA, two basic approaches are possible: Received Signal Strength (RSSI) and Time-of-Flight (ToF).

RSSI is generally useful for distance measurements. However, in the context of DBMA, RSSI is unsuitable. An attacker can increase transmission power at will. Therefore, an attacker is able to artificially decrease distance measurements and appear to be located in the secure area.

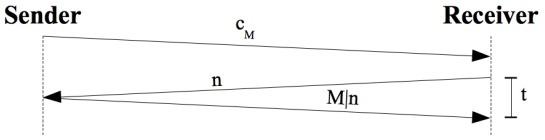

Figure 1. RTTMAP Message Exchange.

ToF measures the propagation delay between nodes. The propagation delay constant is used to derive distances and an attacker is thus not able to artificially decrease distance measurements.

For a simple ToF measurement a sender has to embed a time stamp in the message in order to allow a receiver to determine the signal propagation delay. Furthermore, sender and receiver must have synchronised clocks to an accuracy of a few nanoseconds. Secure clock synchronisation is hard to achieve and, more importantly, the time stamp embedded in the message may not be trustworthy; a different approach to ToF measurement is thus necessary. A Round-Trip-Time (RTT) ToF measurement initiated by the message receiver addresses the outlined issues.

C. Round-Trip-Time Message Authentication Protocol

The aforementioned RTT measurement mechanism must be incorporated into a communication protocol, but it must be initiated by the receiver. RTTMAP described in this paragraph can be used for this purpose. A detailed description and analysis of RTTMAP can be found in [8].

A sender first transmits a control message to the receiver signalling a pending data transmission. The receiver responds with a second control message which, upon reception, triggers immediate transmission of the data message by the sender. Thus, the receiver is able to compute distance from a round trip time measurement using the second control and data messages.

It must be ensured that a sender is not able to transmit the data message early as this would allow him to reduce the distance artificially. Capkun et. al [7] proposed including nonces in ranging messages[1] to avoid early responses and achieve secure RTT measurements. An attacker could respond late, but this is not a problem as the observed distance would increase and he would appear to be outside the secure area.

During network planning, each node is configured with a distance threshold value r that is determined by the deployer. r is set such that all neighbours with which the node shall communicate are closer than r. Nodes that are excluded from communication are assumed to be always further away than r. r obviously has to include any measurement error, as we discuss in Section V.

[1]For secure localisation rather than message authentication.

RTTMAP requires the following three phases for each message exchange (see Figure 1):

1) The sender adds a fresh counter value i to message M. A commitment is computed using the hash function: $c_M = h(M)$. c_M is sent to the receiver.
2) The receiver caches c_M and creates a fresh nonce n. n is sent to the sender as timer is started.
3) The sender returns $M|n$ to the receiver. The receiver takes a timer reading t to calculate distance r'. The receiver recovers M from the response. M is accepted only if $r' < r$, i is fresh, and $c_M = h(M)$.

To avoid attacks on the protocol itself, nodes do not accept messages unless they are in a state where they are expected to arrive. The cached c_M and timer t cannot be changed until the message has been received or a timeout has occurred. RTTMAP uses the nonce to prevent early responses by a potential attacker. Message replay is avoided by using the counter value i. This is especially important as the same message may be sent often and an attacker may be able to form a dictionary of messages and associated hash values. A participant is forced to participate in all three stages of the protocol due to the use of the hash function, so an attacker cannot send a message without participating in the first phase. The hash function is only invoked if the distance measurement is acceptable, thus adding protection.

IV. Implementation

For the implementation we chose the Nanotron DK development kit which comprises an Atmel ATmega128 MCU and a Nanotron NA5TR1 transceiver [1]. The NA5TR1 supports RTT distance measurement and operates in the same power class as commonly used sensor node transceivers such as the Chipcon CC2420. The NA5TR1 uses Chirp Spread Spectrum modulation in the 2.4GHz ISM band. The NA5TR1 has a maximum power utilisation of $33mA$ at $2.7V$, a maximum transmit power of $1mW$ and timing provides ranging accuracy of $1m$ in ideal conditions. Currently no WSN operating system such as TinyOS exists for the platform, so the provided C API was utilised. We used the MIRACL [9] library for the SHA-256 hash function.

A. RTTMAP Implementation Challenges

To implement RTTMAP it must be possible to implement the outlined three phases of a message transmission. This is possible in principal, however, the Nanotron API does not allow us to transmit user data within ranging transmissions. We thus modified the protocol such that it is implementable on the NA5TR1. We call the modified protocol RTTMAP-N. RTTMAP-N is not as efficient as RTTMAP but provides the same properties

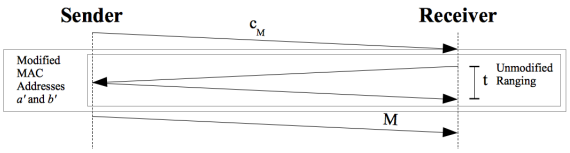

Figure 2. RTTMAP-N Message Exchange.

in terms of participation enforcement. Thus, RTTMAP-N allows DBMA performance evaluation as presented in the next sections.

B. RTTMAP-N Implementation

To achieve RTTMAP equivalent functionality, RTTMAP-N modifies the MAC addresses during the ranging process. Each node is still configured with a distance threshold value r, but four phases for each message transmission are used (see Figure 2):

1) The sender adds a fresh counter value i to message M. A commitment is computed using the hash function: $c_M = h(M)$. c_M is sent to the receiver and the MAC address of the sender is set to a *temporary address* a' based on c_M.
2) The receiver caches c_M and the original address a of the sender. The MAC address of the receiver is *randomised* to b' and a ranging request is sent to a' as the ranging timer is started.
3) The sender sends a ranging acknowledgment to b'.
4) Both nodes restore the original MAC addresses a and b. The sender then sends M to the receiver. The receiver takes a timer reading t to calculate distance r'. The receiver recovers M from the packet. M is accepted only if $c_M = h(M)$, $r' < r$ and i is fresh.

RTTMAP-N does not require a nonce, because the randomised MAC address of the receiver is not known by the sender until the ranging request is received thus providing the same function. RTTMAP-N does not need to insert M into the ranging response because both the delivery of M and the ranging acknowledgment are separately linked to the secure ranging via the hash commitment.

V. Evaluation of DBMA Deployments

To evaluate DBMA performance we collected range measurements from two test deployments (see Figure 3). Deployment A is in a modern office building comprising of earthed metal floor and roof panels with glass office partitions containing metal blinds. Deployment B is a small radio station of traditional breeze block construction with fewer earthed and metallic surfaces. We chose these two deployments due to the different construction techniques and therefore differences in RF propagation

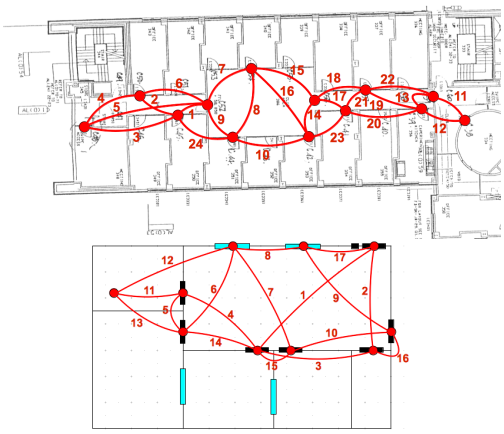

Figure 3. Deployment A (top) and Deployment B (bottom).

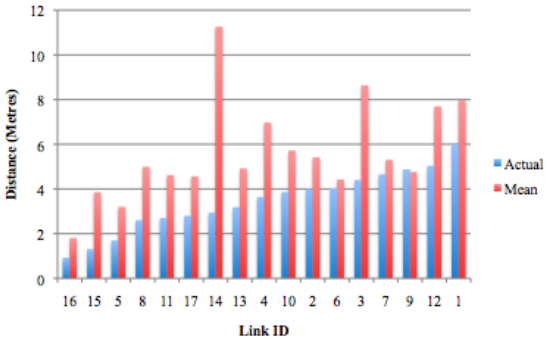

Figure 4. Actual vs. average distance measurement in Deployment B.

patterns. The figures show node positions and communication links available for topology construction.

To conduct measurements, nodes were placed in the intended sensor node positions and 50 RTT measurements were carried out on each available link. These samples give us a picture of the distance measurement distribution on each link. RTT distance measurement distribution is similar for both transmission directions on a link as an RTT measurement requires transmission in both directions. The secure area must be selected to compensate for such measurement errors on used links.

A. Distance Measurement Errors

In an ideal world, we would configure all nodes with a distance threshold such that they can communicate with their neighbours, but not with nodes further away. The security area would then be placed such that it encompassed the distance thresholds for all nodes in the network. In reality, the security area must be enlarged to account for ranging errors. Realistic distance measurements contain two types of errors:

- *Timing Errors:* Timing error is related to the design of hardware and and issues such as clock drift. This

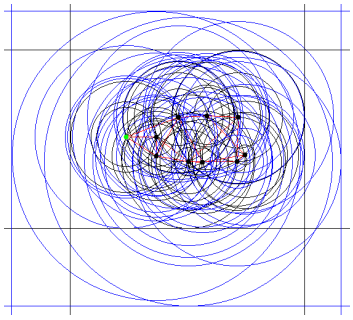

Figure 5. Deployment B. Black circles indicate ideal thresholds. Blue circles indicate required maximum measured thresholds. Black boundary indicates ideal secure area ($339.34m^2$). Blue boundary indicates required secure area ($788m^2$).

error can be negative or positive, but is bounded and only influenced by the receiver of the message (*An attacker is unable to increase this potential negative error*). In the case of the NA5TR1, this results in a distance error of up to $\pm 1m$.

- *Propagation Errors:* Propagation error is related to the signal pathway and edge detection delay in the transceiver. The error is always positive as the signal cannot travel faster than the speed of light. However, the error can be regarded as being unbounded. In our deployments we observe propagation errors of up to $+10m$.

Figure 4 shows the Euclidean distance and the average of measured distance for each link in deployment B. It is clear that there is no reliable correlation between the two values meaning that defining the security boundary requires test data on a per-deployment basis.

B. Secure Area Size

It is not possible to subtract the expected ranging error when measuring communication distance; an attacker might be able to transmit a message to a node with less error by choosing an optimal path. Fitting the secure area to the observed worst-case measurement ensures that an attacker cannot pretend to be located in the secure area, whilst maintaining a high probability that genuine messages will be accepted. Unfortunately this requires that the maximum measurement on each link is used as the authentication threshold, thus enlarging the secure area requirement greatly when links have large errors.

Figure 5 shows the minimum and maximum secure area in deployment B. For practical reasons the secure area has the form of a square and not an arbitrary shaped form. The maximum secure area is more than double the minimum in both deployments. From a practical perspective this results in high financial costs as a lot of land is needed to implement the secure zone. It is therefore desirable to exclude links that increase the secure area requirement undesirably, but are not needed to form a fully connected network. For example, in deployment B it possible to exclude link 14 (see Figure 3) which has huge distance measurement errors (see Figure 4). Nodes could forward messages along link 4 and 5 which have better error values. However, it is not always the case that the exclusion of links with large errors leads to reduction in the secure area size; the location of a link within the network plays a role as well since large errors deep in the network may not require expansion. These topology optimisations are discussed in the next section.

Algorithm 1 Recursive algorithm to find best average value and resulting path to the sink for node n. Where v is a cloned list of nodes visited in this path previously.

```
Function bestAvg( v )
 If v.contains(n) then return null.
 If n is the sink then return {n},0.
 v.add(n).
 Let b = null, c = infinity.
 For each local link l
  Let p,a = l.otherNode().bestAvg(v).
  If p not null
   p.add(n).
   Let s = p.size.
   Let d =(a*(s-1)+l.metric/s).
   If b = null then b = p and c = d.
   If d < c then b = p and c = d.
 Return b,c.
```

VI. Optimising RTTMAP Deployments

In this section we introduce link-pruning to exclude links leading to an unnecessary large secure area overhead. The proposed link pruning algorithm is evaluated using the two deployments outlined in the previous section.

A. Link Pruning Algorithm

To decide which of the available links should be used to form the network topology Algorithm 1 is executed to search all the pathways to the sink from a given node, calculating the average value and selecting the best path. The algorithm is applied to each node in the network and the links on the selected path are enabled. Each link has an associated cost metric which is evaluated in order to decide which links should be included in the network topology. Three different cost metrics were used within Algorithm 1: (1) Physical length of the link; (2) Maximum distance measurement of the link; (3) Overhead contribution.

Metric (1) and (2) are directly available from the deployment. Metric (3) is calculated based on the ideal secure area and signifies the worst case expansion needed in any dimension if the link is enabled, as shown in Figure 6. Thus links with errors that do not require enlargement of the secure zone are not penalised. This approach improves the number of links that can be considered, particularly deeper in the network.

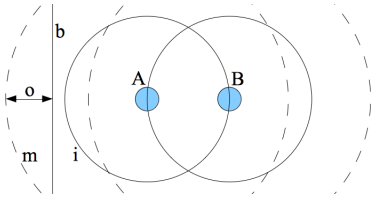

Figure 6. Overhead contribution. For each dimension from each node, the maximum measured distance is subtracted from the distance to the boundary. The worst of all options is used, with a minimum of zero.

	Deployment A			Deployment B		
	O	h	l	O	h	l
No pruning	177.06%	4.08	24	132.22%	3.00	17
Physical (1)	138.41%	6.67	13	125.86%	4.89	9
Meas. (2)	6.23%	5.33	13	1.73%	4.00	10
O. Contr. (3)	3.64%	5.17	13	0.76%	4.89	10

Table I
SECURE AREA OVERHEADS IN DEPLOYMENT A AND DEPLOYMENT B WITHOUT APPLYING ALGORITHM 1 AND AFTER APPLYING ALGORITHM 1 USING METRICS (1), (2) AND (3).

B. Link Pruning Results

To discuss the results we use the term *minimum secure area size* A_{min}, which is the area that would be required in the absence of distance measurement errors; the term *maximum secure area size* A_{max}, which is the area required in the presence of *all* errors and in case *all* available links are used by the network; and the term *required secure area size* A_{req}, which is the area required after link pruning Algorithm 1 has selected links.

To gauge the effectiveness of link pruning we determine the *secure area overhead* O calculated as: $O = (A_{req} - A_{min})/A_{min}$. We also calculate the average hop distance h of nodes from the sink. We also record the number of links l used in the topology after link pruning.

The results are summarised in Table I. In Deployment A $A_{max} = 2598.66m^2$ and $A_{min} = 937.93m^2$ was recorded. In Deployment B $A_{max} = 788m^2$ and $A_{min} = 339.34m^2$ is observed. In both deployments the pruning algorithm using metric (3) achieves the best results: $A_{req} = 972.03m^2$ and $B_{req} = 341.92m^2$.

C. Findings

It is obviously beneficial to use link pruning as it allows us to reduce the required secure area size. In Deployment A the secure area overhead can be reduced from 177.06% to 3.64%, in Deployment B a reduction from 132.22% to 0.76% is achieved. As shown in Table I the average hop distance from each nodes to the sink increases if link pruning is used. Therefore, a decrease in secure area size can be traded for an increase in communication latency.

It has to be noted that the evaluated link pruning algorithm may not be the best available algorithm, we have chosen this method to permit potential use within the network itself. Other link pruning algorithms may be more effective. Link pruning may not be effective in cases where all links exhibit high error; but we have shown this not to be the case in our deployments.

VII. CONCLUSION

We described an implementation of Distance Based Message Authentication for sensor nodes equipped with the NanoLOC TRX NA5TR1 transceiver. As demonstrated, DBMA provides a practical additional layer of defense for WSN deployments.

Using data from real deployments, we evaluated the impact of propagation error on secure area overhead. We proposed link-pruning to reduce secure area size requirements. We show that link-pruning can reduce the required secure area size by a factor of 2 in the investigated deployment scenarios.

Future work will consider the implementation of the original RTTMAP protocol to replace the less efficient RTTMAP-N implementation. To do so it is necessary to change the API provided by NanoLOC. Other aspects are the integration of RTTMAP in low-power MAC protocols and the evaluation of hardware security issues.

REFERENCES

[1] "NanoLOC TRX NA5TR1," Nanotron Technologies, 2008.
[2] "802.15.4 Standard," IEEE, 2007.
[3] Hu, W. and Corke, P. and Shih, W. and Overs, L., "secfleck: A public key technology platform for wireless sensor networks," In proceedings of the 6th European Conference on Wireless Sensor Networks, Cork, Ireland, 2009.
[4] Liu, A. and Ning, P., "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," In Proceedings of the 7th International Conference on Information Processing in Sensor Networks, St. Louis, Missouri, USA, 2008.
[5] Szczechowiak, P. and Oliveira, L. and Scott, M. and Collier, M. and Dahab, R. "NanoECC: Testing the limits of elliptic curve cryptography in sensor networks," In Proceedings of the 5th European Conference on Wireless Sensor Networks, Bologna, Italy, 2008.
[6] Martinovic, I. and Pichota, P., and Schmitt, J. B., "Jamming for good: a fresh approach to authentic communication in WSNs," In Proceedings of the Second ACM Conference on Wireless Network Security, Zurich, Switzerland, 2009.
[7] Capkun, S. and Hubaux, J. P., "Secure positioning in wireless networks," IEEE Journal on Selected Areas in Communications, 2006.
[8] Chung, T. and Roedig, U., "On The Feasibility of a New Defense Layer for Wireless Sensor Networks using RF Ranging," In proceedings of the IFIP Network and Service Security Conference, Paris, France, 2009.
[9] "MIRACL," Shamus Software, 2009.