On The Feasibility of a New Defense Layer for Wireless Sensor Networks using RF Ranging

Tony Chung and Utz Roedig
Infolab21, Lancaster University, UK
Email: {{a.chung|u.roedig}@lancaster.ac.uk}

***Abstract*—Cryptography is commonly used to provide link-layer message authentication in wireless sensor networks. However, keys are susceptible to compromise and introduce management requirements. Avoiding keys can therefore deliver security and management benefits. Our paper introduces and discusses the feasibility of RTTMAP, a protocol that uses radio frequency ranging for message authentication. RTTMAP uses secure round-trip-time with hash functions to determine the minimum distance of a transmitter. Transmissions from outside of a defined radius are rejected without requiring keys. We provide our motivation, an evaluation of our findings and continuing research challenges. We find RTTMAP offers higher security, costs about twice the energy of keyed message authentication but complicates MAC protocol selection. *[1]**

I. Introduction and Motivation

Wireless Sensor Networks (WSN) provide a sensing platform for a variety of sensing applications and consist of a number of nodes, each with wireless communication and an independant power source (i.e. batteries). The lack of wired infrastructure and inherent redundancy offer benefits for sensing applications. These benefits include easier management, lower deployment costs, better scalability and improved resilience to failure. However, WSN design is deliberately minimalist to support long battery life and reduced financial cost.

WSN applications are now emerging in scenarios where there is considerable risk to life, the environment or finance if malicious messages are introduced or network operation is disrupted. Due to the aforementioned design limitations, protocols providing WSN security must not only provide for confidentiality and authentication, but also availability, survivability and graceful degradation.

Current approaches to avoid malicious use of the network are mainly based on cryptographic protection. These seek to protect the privacy and integrity of messages. Modern communication protocols such as IEEE 802.15.4 support this at the link-layer, thus providing access control. We identify three problems with cryptographic approaches.

Firstly, since a WSN is a multi-hop network, link-layer cryptographic protocols provide good protection only if all the keys remain secure. Once a single key has been compromised, it can be used to inject messages into the network. Network security thus degrades badly when keys are obtained by an attacker. Avoiding this requires hardening of nodes and regular key replacement, both of which can be challenging. Thus, we are motivated to supplement or replace these protocols with protocols that do not use keys at all.

Secondly, denial-of-service attacks against security protocols themselves are more serious in a WSN due to the minimalist nature of the nodes. Such attacks vary, but an attacker will aim to deplete resources with as little effort as possible. One attack is the consumption of resources carrying out message authentication; even if a message is found to be malicious, the cryptographic algorithm still has to be executed. If a public key algorithm is used, this can be very serious. In others, an attacker might aim to flood a node's communication buffer. We are therefore interested in methods which can handle denial-of-service.

Finally, key management in a sensor network is not a straightforward task since it requires the secure generation and distribution of keys for every pair of adjacent nodes. Avoiding this overhead and the associated security risks is beneficial.

Cryptography is not the only approach to message authentication available in a WSN. This paper explores the benefits and challenges of using Radio Frequency (RF) ranging to authenticate messages in WSNs. RF ranging can offer accurate distance information that can be very hard for an attacker to forge. It is thus an ideal security parameter for authentication purposes in some scenarios.

We introduce RTTMAP, a method of using RF ranging to authenticate a transmitter based on distance. RTTMAP integrates secure round-trip-time measurement into message transfers at the link-layer and enforces a maximum permitted distance radius. New communication standards and hardware are now emerging which support the required ranging functions. We believe that RTTMAP is a potential authentication solution in scenarios where access to a specific physical area is restricted and since it does not use keys, it is not vulnerable to their loss. We evaluate the security benefits, overheads and upcoming research challenges of RTTMAP via analysis and simulation of our planned node design.

Our paper is organised as follows. Section II introduces our scenario and related motivation. Section III explores related work in the field of WSN security and RF localisation. Section IV introduces RTTMAP and related principles. Section V introduces our planned node design, protocol and simulated implementation. Section VI evaluates the performance of the scheme, its cost under attack and implications for MAC protocol selection. Finally, in Section VII we conclude with future work.

[1]NB: A version of this paper on IEEE Xplore contains incorrect calculations in Section 6. This version of the paper contains corrected calculations.

II. Scenario

We are implementing a building intrusion application based on a WSN comprising of a number of modified low-power Tmote Sky[13] sensor nodes. Each sensor node is equipped with security sensors and sends observation reports to a controller station. Messages travel across multiple nodes in the network. Although most messages are intended for the controller, parts of the system may still require in-network communication thus requiring link-layer security.

Since the network needs to be left running for a long period of time, and some nodes are hidden or hard to reach, there is also a motivation to extend battery life for as long as possible. However, if link-layer security is compromised, problems such as denial-of-service may arise that drain these resources.

To avoid introduction of transmitters and other electronics, all persons using the building can be checked for devices on entry and exit; but there remains the risk of cryptographic key material being stolen by untrustworthy visitors.

The WSN is not the only security system in use, but it is important for it to continue operating even when under active attack. The building is sufficiently large to permit rerouting of messages should a particular area come under attack. By using end-to-end cryptographic authentication (like in [1]), even if some keys are stolen, the WSN can continue to operate in a degraded state to deliver data to security personnel.

We were motivated to investigate an authentication mechanism that can detect external intrusion without keys. Ranging is an ideal option, if sufficiently accurate, as each node can be set with a simple parameter (the maximum distance, set to the nearest security boundary) and reject messages sent from outside that distance.

III. Related Work

Security in sensor networks is a very active research area mainly focusing on cryptographic protocols. Sensor network applications often omit or use weak security; we believe this is partly motivated by difficulties in integrating security protocols and managing relevant keys. RTTMAP helps to avoid this problem by (1) existing entirely below the link-layer and (2) not requiring keys and therefore key management.

Link-layer cryptography is motivated by the need for many sensor network applications to implement in-network processing such as aggregation and filtering. (End-to-end cryptography is simply incompatible with these applications since neighbouring nodes are unable to view or modify data that passes through them.) Examples include SNEP[5], TinySec[6] and transceiver-based security found in IEEE 802.15.4. The greatest weakness of link-layer cryptography is key compromise (theft or cryptanalysis). The security of the overall network degrades badly once this occurs since messages are given a new signature on each hop. This motivates the use of end-to-end cryptography in some deployments[1]. A replacement, or additional layer of protection, is thus desirable.

Broadcast authentication is used to avoid injection of malicious queries and code updates. Examples include μTESLA[5] and Secured Deluge[7]. We note that all of these protocols exhibit some form of denial-of-service vulnerability, such as energy depletion attack in public-key based methods or buffer attacks in time-delayed schemes such as μTESLA. These schemes would benefit from an additional layer of security as a defence. Although RTTMAP does not currently support broadcast, in some scenarios adapted broadcast protocols could be used with RTTMAP.

Security at the physical layer has been shown to provide enhanced security. Ahmed et al[2] integrates keys with error correction codes. By applying security at this layer, it is harder for an attacker to deduce the headers in a message, let alone successfully deliver one to another node. If a message cannot be delivered, it is harder (or infeasible) for an attacker to attack cryptography in the higher layers. RTTMAP takes a similar, physical layer, approach.

Secure ranging is an established WSN research area for positioning applications[8]. A number of proposals exist which use signal strength (RSSI), ultrasound[9] or RF timing[10] to securely measure node seperation. RSSI is widely regarded as unreliable[11] and requires special attention[3] to avoid potential power spoofing attacks. Ultra-sound uses a different, and expensive, communications medium and prevents fully RF-based message transmission.

Chirp-Spread-Spectrum (CSS) and Ultra-wideband (UWB) have recently been introduced in the newer IEEE 802.15.4a standard for use in wireless sensor networks. These technologies offer a number of benefits that improve radio ranging. New transceivers have recently emerged[4] that provide built in ranging as part of message transmission. Although these do not properly fit our scheme, they do prove that high accuracy RF ranging is now feasible in low power devices.

IV. Scheme

A. Background

Obtaining range measurements using the radio tranceiver eliminates the requirement for additional security hardware[12] and allows transmissions to be coupled with ranging. Ranging methods based on signal strength (RSSI) should not be considered for security purposes because an attacker might modify power output to manipulate measurements.

Another method, time-of-flight (ToF), times the delay between message transmission and message reception to obtain the range. ToF requires that the clocks of both nodes are synchronised, which is difficult to achieve securely and with sufficient accuracy. ToF also requires that the sender include a timestamp in each message, which therefore requires trust. ToF can therefore be manipulated.

The same principle as ToF can be extended to measure the Round-Trip-Time (RTT). In RTT a message is sent to, and returned by, the other node. A timer is started when a message is transmitted and then stopped when it has been received. The delay is taken directly from the timer, thus RTT does not require timestamps or clock synchronisation. We thus consider RTT, as it is harder to manipulate.

RTT and ToF have an interesting property in that they operate using a fixed propogation speed, the speed of light, that cannot be accelerated by an adversary using current technology. Therefore, in RTT, if we eliminate the potential for the other node to respond early, we can enforce a truth boundary. If the other node cannot reply early, it cannot appear to be closer, although it can reply late and thus appear further away. We are not concerned with distance exaggeration, but we do exploit the former.

This principle has been applied before[10], but not for message authentication, and requires that the request message contains a value (or 'nonce') that cannot be predicted by the node that returns the message. Therefore the other node cannot feasibly reply early.

B. Round-Trip-Time Message Authentication Protocol

Secure RTT must be initiated from the secure side of the link; therefore if it is to authenticate a transmitter it will require an additional two messages per transfer[2] event. It is also important that the ranging takes place with the same node that sent the trigger message. We now outline our protocol, RTTMAP (Figures 1 and 2) and explain its security properties.

Each node must be configured with a maximum communication range r. r is the shortest distance to the security boundary where attackers can transmit into the network. Note that r must be calculated to take into account timing inaccuracies, modulation duration, and the turnaround time on Node A (at the start of Phase 3). When configured, an attacker will not be able to respond quickly enough to reduce the ranging measurements to below r.

There are then three phases for each message transfer, shown in Figure 1:

1) The sender (A), prepares message M and applies hash function $h()$ to produce commitment c_M. c_M is sent to the receiver (B). M contains a counter i to avoid opportunistic replay attacks.
2) Node B caches c_M and generates nonce n. n is sent to Node A and timer t is started.
3) Node A sends $M|n$ to Node B. Node B recovers nonce n' and M. M is accepted only if $n' == n$, $c_M == h(M)$, t is within range and i is unique.

Nodes ignore unsolicited messages received during phase 3 and the cache entries and timer functionality time out after a permitted period. These measures avoid denial-of-service attacks on the protocol itself. The timer and cached C_M cannot be reset until they have expired or been used.

Figure 2 shows an example of a small deployment of nodes using RTTMAP.

C. RTTMAP Security Objectives

The use of Secure RTT prevents an attacker from pretending to be within the acceptable security range. However, it is necessary to add additional protection to avoid hijack attacks on existing sessions.

RTTMAP packets include a counter value i which avoids opportunistic replay. This is important if the same report may be sent often as it reduces the chance of an attacker guessing message content, and thus hash values, in advance.

RTTMAP also sends a one-way hash c, rather than the message M itself in the first phase. This forces the attacker to participate in all three stages. The one-way property of $h()$ prevents generation of a valid message within the limited time frame and without knowledge of M and attacker cannot generate c in advance.

RTTMAP does not reset the timer or update the cached committment c_M to avoid attacks which may involve an attacker repeating his initial transmission of $h()$ in hope that the timer value on the receiver will be continually reset (and thus resulting in a smaller value).

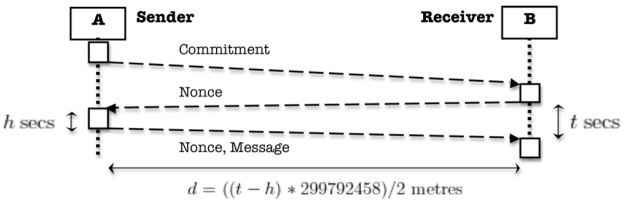

Figure 1. RTTMAP involves three message exchanges. (1) A committment. (2) A nonce. (3) The nonce and a message. The distance d is derived from the time t minus the response overhead h as shown.

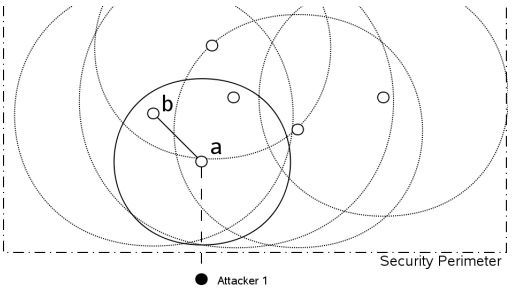

Figure 2. A maximum distance is set for each node (circles) to allows internal communication only. An attacker cannot inject from the outside (dashed line to a), but internal communication can occur (solid line to b).

V. Implementation

A. Node Design

Our evaluation is based on our planned integratation of a Chirp Spread Spectrum (CSS) transceiver (the NA5TR1 from Nanotron) with a low-power CPU (MSP430). This will result in an architecture similar to the Tmote Sky[13] sensor node. The NA5TR1 is different from the existing radio due to it's use of CSS rather than older techniques in the CC2420. Although the NA5TR1 does not properly match RTTMAP in functionality[3], we can use it as a reference for our feasibility

[2]To avoid ambiguity, we use the term *transfer* to refer to the transfer of data from one node to another. Notice that in RTTMAP this involves three message transmissions.

[3]The *Delivery* message would need to be cached and updated within the transceiver itself, in order to permit fast turnaround on receipt of a *Probe* message. We plan to address this later in our future research.

RTTMAP Init

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Source															
Destination															
Type								Commit							

RTTMAP Probe

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Source															
Destination															
Type								Nonce							

RTTMAP Delivery

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Source															
Destination															
Type								Nonce							
								Length							
Data*															
CRC															

Figure 3. Frame structures for RTTMAP.

evaluation as it implements RF RTT ranging.

B. Frame Structures

RTTMAP uses three frame structures, one for each phase of the protocol. See Figure 3. These are identical regardless of the MAC protocol. All structures include 2 byte *source* and *destination* addresses with a 1 byte *type* field used to differentiate the frames.

Init messages, in Phase 1, initialise a transfer and require one field. The *commit* field (4 bytes) contains the truncated output c_M of hash function $h()$. The total length is 9 bytes.

Probe messages, in Phase 2, begin range measurement and require one field. The *nonce* field is 4 bytes long. The total length is thus 9 bytes.

Delivery messages, in Phase 3, complete ranging and transfer using four fields. The *nonce* field is 4 bytes long, containing the nonce delivered by a *Probe*. The 1 byte *length* field contains the *payload* length (maximum of 255 bytes). The total length is therefore between 11 and 266 bytes.

C. MAC Protocols

RTTMAP involves close interaction between lower layers in the network stack. This is due to the raw access required to perform ranging measurements using the transceiver. In order to properly evaluate RTTMAP, it is necessary to specify MAC protocols to explore the consequences of these choices.

The *CSMA* protocol checks the channel before transmitting *Init* and *Probe* messages. If the channel is busy, there is a random backoff before further attempts. Because of the need to carry out RTT measurement, no channel assessment is carried out before sending *Delivery* messages.

The *duty-cycling* protocol is a derivative of TinyOS Low Power Listening (LPL). LPL disables the radio, except during defined periods in an epoch. This saves energy, but requires that transmissions be extended so that they can be delivered without the need for time synchronisation. Channel assessment is carried out in the same way as the CSMA protocol.

The *TDMA* protocol uses loose[4] time synchronisation. Nodes transmit only in their own time slot in each epoch. The slots are long enough to complete a message exchange in RTTMAP. Channel assessment is not used.

[4]Although TDMA employs node synchronisation, in reality it is insufficiently accurate to support ToF range measurement.

D. Node Configuration

All nodes require an address and a maximum range r. These are computed in advance and stored before deployment. All nodes require a hash function, we use SHA-256 as efficient 16-bit implementations (i.e. MIRACL) are available.

During operation, each node will need a timer, operating at a minimum of approximately 500MHz, to acheive a RTT accuracy of approximately 33cm. A cache of 8 bytes is required for the committment and nonce during each exchange. We expect that these features will eventually be implemented within the transceiver to eliminate ranging error.

E. Simulation

We simulate RTTMAP in our own Java WSN simulator called sensorsim. Sensorsim is a state-based program that simulates a physical space containing sensor nodes and active transmissions. By simulating network state changes at a scale of nanoseconds, it is possible to implement working RTTMAP with high accuracy. We optimise sensorsim by eliminating as many state updates as possible, it is not feasible to simulate every nanosecond. Simulated node and transmission objects are scanned to skip redundant updates thus reducing runtime.

All objects are polled on each iteration, which ensures updates occur in lock step. During the update it is possible to change various parameters. For example, the simulator can change energy parameters to mimick power-cycling protocols or can calculate the current radius and signal strength of a transmission (informing relevant sensor node objects of message arrival). By carefull design we can simulate a network containing an arbitary number of nodes running various protocols. We can thus gather energy data and count transmissions.

We used datasheets[4] of popular WSN hardware, like the MSP430 and NA5TR1 transceiver, to derive energy values for various states. We combined this with cryptographic performance measurements taken from Tmote Sky nodes. The simulator does not provide real-world propogation characteristics (like reflection), but can provide useful proof-of-concept data.

VI. Evaluation

RTTMAP increases the number of message transmissions three-fold. Depending on the Medium Access Control (MAC) protocol, this causes additional overhead compared to other schemes and causes interaction resulting in changes to network reliability. We investigate our initial findings here.

A. Energy Overhead

There are two elements to energy overhead in a transfer, the cost of transmission and the cost of cryptography. These should take into account for both sender and receiver. All our measurements are in milliamp seconds (mAs), which is a second of current drain at one milliamp. Our nodes run at 2.3 Volts, the MSP430 draws 1.9mA and the NA5TR1 draws 30mA if sending or 34mA if receiving.

1) Communication Overhead: Since the NA5TR1 draws less current when transmitting compared to receiving, in applications where there is no duty-cycling of the radio, there is no extra energy penalty[5] in transmission. Thus RTTMAP protects against attacks on cryptographic functions with no large overhead (although channel contention may become an issue, see below).

In reality, low-power applications will duty-cycle the radio power to save energy. In low-power state the radio is not able to receive, so MAC protocols have to ensure that messages can be successfully exchanged. Two approaches are common, time synchronisation and extended transmission.

Time synchronisation protocols aim to wake all transceivers at the same time, so that all transmissions can be heard. The synchronisation does not need to be accurate enough for RF ToF, but is needed to avoid clock drift. Once the synchronisation cost has been accounted for, the cost of using the radio has an upper bound for the duration of deployment (roughly equal to the duty-cycle multiplied by receive cost).

Extended transmissions are used by protocols to avoid synchronisation. These extend transmission so that nodes will always hear part of the message and stay awake to hear the end of the message. The cost of transmitting a message is therefore the cost of keeping the radio active for a full epoch. In a duty-cycled MAC protocol with an epoch length of e ms, the cost of sending a message is $c{*}e/1000$ mAs, where c is the cost in milliamps to transmit a message. This cost should be doubled since receivers will need to stay awake for longer. In TinyOS LPL (2% duty-cycle with a 550ms epoch length) this results in a transfer cost of 33 mAs ($2{*}16.5$) for *Init* and *Probe* messages. The length of the message is unimportant provided it fits within the epoch. *Delivery* message cost is based on the length of the message, since both radios are awake and no extension is required.

Each message sent using an NA5TR1, at a speed of 1mbps, is encoded into chirps of 1μs duration per bit. We assume a preamble, sync word and tail element totaling approx 70μs. Thus *Delivery* messages take between 158μs to 2198μs of time, costing between 0.00474mAs to 0.06594mAs. This cost must also be multipied by two.

2) Cryptographic Overhead: RTTMAP requires two invokations of a hash function, in our case SHA-256. From our experiments with the MIRACL[14] library on the MSP430 (current 1.9mA) we have calculated that SHA-256 takes 10.17ms to initialise, followed by an approximate time of 0.032ms for each byte hashed. Thus the minimum cost of SHA-256 is 10.17ms and the maximum cost is 18.33ms to protect a full 255byte payload. The total minimum cost is therefore 0.038646mAs (based on 20.34ms) and the maximum is 0.069654mAs (based on 36.66ms).

By contrast, a regular keyed authentication protocol is likely to require two invokations of a signature function. We evaluate AES-256 in CBC-MAC mode. On our Tmote Sky nodes, this completes a single block encryption (of 16 bytes) in 1.11ms, each byte thus requires approximately 0.069ms of CPU time. The minimum cost of AES-256 is thus 0ms and the maximum is 17.664ms to protect 256 bytes (although the maximum payload is 255 bytes, AES operates in 16 byte blocks). Thus, 0ms is the minimum cost and 0.0671232mAs the maximum (based on $2 * 17.664$ms).

RTTMAP is thus more expensive in computation for each exchange in some situations, but we have not considered public-key algorithms which can be used at the link-layer and are considerably more expensive (taking several seconds at best). These can be found in some broadcast authentication schemes and one of our aims is to protect these.

3) Total Overhead: Performing a transfer between two nodes using RTTMAP thus costs a minimum of 0.038646mAs for a 0 byte message in a non-duty cycled MAC to a maximum of 66.201534mAs for a full size message in the LPL case. This is based on the three transmissions plus the cost of two invokations of the hash function SHA256.

We compare this against a traditional approach using Message Authentication Codes (MACs). Protocols such as TinySec[6] require calculation of a 4 byte MAC (costing between 0mAs and 0.0671232mAs). Although they extend the packet length, this extra length is absorbed either because transmission is free or an extended transmission is used. Thus existing methods can cost between 33 and 33.0671232mAs.

As we see, RTTMAP costs just over 50% extra in this case. We argue that this investment is worthwhile to avoid keys.

B. Energy Performance Under Attack

Before considering RTTMAP as a countermeasure against denial-of-service attack, it is important to explore the cost of RTTMAP when it is attacked itself. We show that the worst case direct energy attack possible by an adversary wastes the equivallent of one full epoch message transmission plus one full epoch message reception (total: 33mAs).

Adversaries that inject an *Init* message will cause the return of a *Probe* message. The caching of the commitment and timer operation will cease after a defined period. The other problem is the possibility of blocking network access by injecting continual *Init* messages and flooding the cache, we argue that the attacker would need to jam to frequency to carry out this attack since the timeout is so short.

Injection of *Probe* messages will cause the return of a *Delivery* message if one is waiting. Although the node will ignore clear channel assessment when replying, we argue that the attacker is probably able to disrupt the network anyway by communicating with that node. Since the *Init* and *Probe* messages have no network payload, they cannot be used to flood the network - which is a more serious concern.

Injection of a *Delivery* message will result in a number of operations to authenticate the payload. This part of the protocol has been deliberately designed for these attacks. The severity of such injections depends on how far the process reaches. The first two tests determine if the *Delivery* is

[5]We have not considered the cost of message transfer between the CPU and radio.

expected and if the nonces match. These comprise of simple instructions and are thus inexpensive.

The node might then carry out a hash operation on the malicious message, but this is cheaper than a message transmission.

Only if the message passes all these operations will the message then be transmitted to the next hop (and onto its destination). Considerable effort and luck is required by the attacker. Unlike key compromise, the attacker will need to restart the attack to repeat the attack. Keyed protocols would now be broken. This is a strength of RTTMAP.

C. Interation with MAC Protocols and Network Reliability

RTTMAP closely interacts with the MAC protocol as transmission of *Delivery* messages cannot be delayed by MAC functions and each transfer involves three transmissions. The implications depend on the type of MAC protocol employed.

In the TDMA style protocol, the main concern is slot length. The slot length must be chosen such that it can accommodate the propogation delay of the three transmissions. Thus the slot length is primarily related to the distance between nodes and the modulation duration. The maximum frame sizes are 9, 9 and 266 bytes. These translate into modulation durations of 142μs, 142μs and 2198μs when the 70μs overhead of preambles is considered. The total delay purely in modulation is thus 2482μs. Assuming a maximum range of 300 metres, we need to add approximately 3μs for radio propogation time (as there are 3 exchanges). Thus the shortest TDMA slot length is approximately 2.5ms. This would allow approximately 40 slots per epoch of 100ms, which we feel is quite acceptable.

In a CSMA style protocol, channel contention and reliability are directly affected by RTTMAP. Message transmissions effectively take longer, and they affect a wider area. The lack of channel assessment upon transmission of *Delivery* messages means that there is a greater possibility of collision. Since *Delivery* messages are the largest, this poses big problems with CSMA. Channel assessment could be used, but then the message would then be dropped by the receiver for being late.

To validate this theory, we simulated networks of CSMA-based sensor nodes randomly placed in a 100x100x100 metre area. Each sent a status report to the sink (also randomly placed) at random intervals (of between 100ms and 1 second). In order to avoid unrealistic synchronisation at startup, all nodes started transmitting at random offsets. The simulation ran for 1 simulated second in each case. The experiment was repeated 20 times for each network size, at the end of the simulation the number of transmissions sent and successfully arrived were recorded and converted into reliability.

We can observe in the results (see Figure 4) that reliability gradually degrades as the network becomes denser when using CSMA, thus we find that RTTMAP works best in a TDMA style protocol, because TDMA does not require channel assessment and therefore collision is less likely. Obviously such an approach would require a sufficient number of time slots to maintain high network availability, as we have already shown it is possible for 40 nodes to share an epoch of 100ms, affording each node 10 transmission opportunities each second.

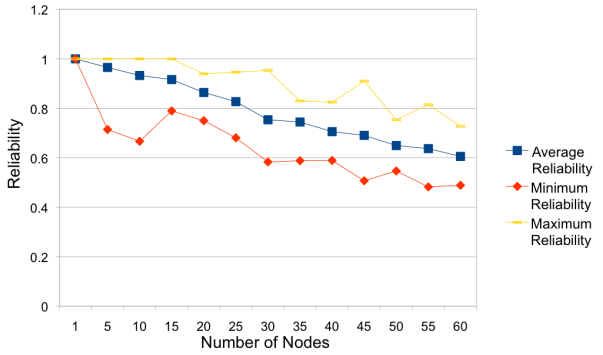

Figure 4. CSMA reliability of randomly placed nodes and random transmission intervals in a 100x100x100 metre area.

VII. Conclusion and Future Work

This paper has introduced RTTMAP, a message authentication protocol based on secure round-trip-time measurements. We have shown that RTTMAP eliminates the need for keys in some scenarios by restricting access to nodes within a defined security boundary. We find that RTTMAP costs no extra in energy overhead where no duty-cycling is in place, or about 2 times the overhead where duty-cycling is in place. On simulation of RTTMAP, we find that CSMA is a poor choice due to the difficulties in handling channel assessment. Thus we find that TDMA is the best choice for MAC with RTTMAP.

Implementation of RTTMAP on real hardware is the next phase of our research. We will investigate the impact of real world interference, realistic propogation and further security enhancements. We note that RTTMAP could also be applied to conventional wireless networking, so our work will involve both WSN and conventional networks.

References

[1] A. Chung, and U. Roedig, “DHB-KEY: An Efficient Key Distribution Scheme for Wireless Sensor Networks,” 2008.

[2] A. Ahmad, A. Biri and H. Afifi, “Study of a new physical layer encryption concept,” 2008.

[3] J. Yang, Y. Chen and W. Trappe, “Detecting Sybil Attacks in Wireless and Sensor Networks Using Cluster Analysis,” 2008.

[4] Nanotron Technologies, “nanoLoc TRX Transceiver (NA5TR1) Datasheet,”, 2008.

[5] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, “SPINS: Security Protocols for Sensor Networks,” 2001.

[6] C. Karlof, N. Sastry, and D. Wagner, “TinySec: a link layer security architecture for wireless sensor networks,” 2004.

[7] P. Dutta, J. Hui, D. Chu, and D. Culler, “Securing the deluge Network programming system,” 2006.

[8] A. Srinivasan and J. Wu, “A Survey on Secure Localization in Wireless Sensor Networks,” 2008.

[9] R. Mayrhofer and H. Gellersen, "On the security of ultrasound as out-of-band channel," 2007.

[10] S. Capkun and J. Hubaux, “Secure positioning of wireless devices with application to sensor networks,” 2005.

[11] K. Muthukrishnan, M. Lijding, N. Meratnia, and P. Havinga, “Sensing motion using spectral and spatial analysis of WLAN RSSI,” 2007.

[12] W. Hu, P. Corke, W. Shih, and L. Overs, “secFleck: A Public Key Technology Platform for Wireless Sensor Networks,” 2009.

[13] Sentilla Corporation, “Tmote Sky Low Power Wireless Sensor Module Datasheet,” 2006.

[14] Shamus Software Ltd., “Multiprecision Integer and Rational Arithmetic C/C++ Library,” 2008.