Spring 2015

Sensor authentication in collaborating sensor networks

Jake Uriah Bielefeldt

SENSOR AUTHENTICATION IN COLLABORATING SENSOR NETWORKS

by

JAKE URIAH BIELEFELDT

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2014

Approved by

Sriram Chellappan, Advisor
Wei Jiang
Dan Lin

ABSTRACT

In this thesis, we address a new security problem in the realm of collaborating sensor networks. By collaborating sensor networks, we refer to the networks of sensor networks collaborating on a mission, with each sensor network is independently owned and operated by separate entities. Such networks are practical where a number of independent entities can deploy their own sensor networks in multi-national, commercial, and environmental scenarios, and some of these networks will integrate complementary functionalities for a mission. In the scenario, we address an authentication problem wherein the goal is for the Operator O^i of Sensor Network S^i to correctly determine the number of active sensors in Network S^i. Such a problem is challenging in collaborating sensor networks where other sensor networks, despite showing an intent to collaborate, may not be completely trustworthy and could compromise the authentication process. We propose two authentication protocols to address this problem. Our protocols rely on Physically Unclonable Functions, which are a hardware based authentication primitive exploiting inherent randomness in circuit fabrication. Our protocols are light-weight, energy efficient, and highly secure against a number of attacks. To the best of our knowledge, ours is the first to addresses a practical security problem in collaborating sensor networks.

ACKNOWLEDGMENTS

I would like to thank my advisor for his help and direction in this project. Without it, this undertaking would have stalled long ago.

I also wish to thank my committee for taking the time to read through this and for providing additional help in satisfying my thesis requirements.

Finally, I want to thank my family, for supporting and encouraging me to see this to the end.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

NOMENCLATURE

Symbol	Description
O^i	Operator of Sensor Network S^i
s_k^i	Sensor s_k belonging to the i^{th} Sensor Network
r	Query round
$N_{r,k}^i$	Nonce shared between sensor s_k^i and O^i for Round r
Y_k^i, Z_k^i	Two secret keys shared between Sensor s_k^i and O^i
PUF_k^i	Physically Unclonable Function of Sensor s_k^i
$C_{(r,k)}^i$	Challenge Vector for Sensor s_k^i in Round r
$A_{(r,k)}^i$	Authentication Challenge for Sensor s_k^i in Round r
$PUF_k^i(C_{(r,k)}^i)$	PUF response of Sensor s_k^i to Challenge $C_{(r,k)}^i$ in Round r

1. INTRODUCTION

1.1. INTRODUCTION TO PROBLEM

Wireless Sensor Networks are proving to be indispensable technologies in many military and civilian settings. Practical necessities today both in military and civilian scenarios indicate that sensor networks in the near future will not be operating entirely independently, but will rather collaborate with peer networks owned and operated by other entities to collaborate on mission tasks. However, when missions involve multiple countries and/or commercial perspectives, complete trust between collaborating networks is not practical. Consider the following two scenarios:

1.1.1. Multi-Country Scenario. There is an abundant amount of natural phenomena that can occur in which several countries are affected. Earthquakes can affect numerous regions across multiple countries, volcanic debris can cover hundreds of square miles, and tsunamis can reach entire coastlines. Detection of these events in order to provide advance warning and aid is significantly important to all the countries vulnerable to such a disaster, and by collaborating with nearby countries, larger sensor nets can be deployed to detect such phenomena as they form and occur at further distances. However, complete trust is improbable as each country will still also possess goals and agendas that may not necessarily be advantageous to the other collaborating countries (pollution and climate policies, etc.).

1.1.2. A Commercial/Environmental Scenario. With commercial and environmental applications of sensor networks like soil monitoring, weather prediction, healthcare, etc., becoming feasible, there is an interest today in sensor-clouds [1, 2, 3, 4] where multiple independent sensor networks are integrated into a cloud framework providing services not possible with a single sensor network. It is likely that individual networks, from competing businesses and organizations may compromise overall functionality of the integrated network and services for selfish gains.

1.2. PROBLEM ADDRESSED

In this thesis, we address the following problem - Given *n* collaborating S^1, S^2, S^3, …, S^n, how can the Operator O^i of Network S^i correctly authenticate active sensors in its network.

This problem is clearly unique to scenarios where multiple sensor networks collaborate, and is practical, since knowing which are active (i.e., functioning) sensors in its own network is critical for network operators. Note here that the solution to this problem is not trivial in the presence of other untrusted sensor networks. When Operator O^i of Network S^i issues a query requesting sensors that are active in its network to report, sensors in another network S^j can masquerade as sensors in Network S^i, packets can be dropped, corrupted, or replayed during forwarding, and malicious entities may also fake O^i.

1.3. OUR CONTRIBUTIONS

We propose two handshaking protocols to solve the above problem in this thesis. Our protocols rely on Physically Unclonable Functions (PUFs). PUFs are circuits in hardware that provide hardware based authentication of a device. Briefly, given a challenge, a PUF circuit generates a verifiable response. The salient feature of the PUF design is that since their behavior is based on inherent randomness of physical hardware during fabrication, their behavior is not predictable before hand, nor is the behavior clonable. Depending on the hardware characteristics and physical property exploited like circuit delays, voltage values at power-up, ring oscillator frequencies, PUFs have been designed with a large number of challenge response pairs up to 2^{64} with minimal increases in circuit overhead and latency [5]. Our protocols use a combination of PUF responses, XOR encryption and aggregation to address the authentication problem, while being resilient to a variety of attacks.

2. PRELIMINARIES

2.1. PRELIMINARIES OUTLINE

In this section, we present important preliminaries related to our authentication problem and proposed protocols. In Section 2.1, we present the overall system model. The problem formulation is presented in Section 2.2. Section 2.3 discusses a number of attacks compromising the authentication problem. A brief overview of Physically Unclonable Functions, which form the core technology used in our authentication protocols, is presented next in Section 2.4.

2.2. SYSTEM MODEL

In this thesis, we are concerned with a network of independently operated but collaborating sensor networks. Figure 1 illustrates a simple case, where there are three sensor networks collaborating in a deployment field. Let us denote these sensor networks as S^1, S^2, S^3. For illustration, let us assume that S^1 is a network of temperature sensors, S^2 is a network of infra-red sensors, and S^3 is a network of seismic sensors. These three sensor networks are independently owned and operated by O^1, O^2, and O^3 respectively, and are expected to collaborate on the field, and communicate with each other. A practical application in this scenario is intruder sensing via fusing information from multiple sensors in multiple networks, despite each sensor network independently executing its own mission.

All sensors are assumed to be static. A sensor in one network may use sensors in another network during routing. A sensor in one network may or not be interested in the information communicated by a sensor in another network. There is some key management scheme that is used by the sensors to protect their communications from eavesdropping by external adversaries. Since sensors can be faulty/ fail/ or be energy depleted, the number of active sensors in any network can change over time. Because of the collaborative nature of the sensor networks, each one is assumed to be able to read to some extent the messages sent by another sensor network.

2.3. PROBLEM FORMULATION

The problem we address in the above system model is the following. How can Operator O^i correctly determine which are the active sensors in its own Network S^i whenever it wishes to. We can see from Figure 1, the number of sensors in Networks S^1, S^2, and S^3 are 10, 12, and 8 respectively. However, as time goes on, sensors in a particular network may become faulty, may fail, or may be become energy depleted. If a significant number of sensors in a particular Network S^i does become in-active, Operator O^i may desire to know this so that corrective action can be subsequently taken to mitigate network deficiency. Note that in practice, such a query from O^i will not arrive very often. It is expected to be generated over longer time intervals, or when O^i suspects any major change in the network state.

2.4. ATTACKS COMPROMISING AUTHENTICATION

Because of the collaboration with other sensor networks, there are two different sets of potential adversaries. The first type of adversary is the one who is external to all of the sensor networks. The second type of adversary is one that is part of the collaborative sensor network. This is an adversary who is also a friend, one whom a primary goal of accomplishing a given mission is shared, but there may also be a secondary goal of denying some amount of information to their collaborating partner or to learn secrets that were not meant to be shared.

The external adversary can easily launch eavesdropping attacks at the sensor network in an attempt to learn secrets and vulnerabilities of the sensors and the network. They can also use masquerade and reflection attacks to trick sensors and / or the operator into revealing secrets and responses used in the authentication process. This attacker can also launch DOS, jamming, and routing attacks in an attempt to disrupt and deny communications in the sensor network. Finally, this attacker can attempt to physically

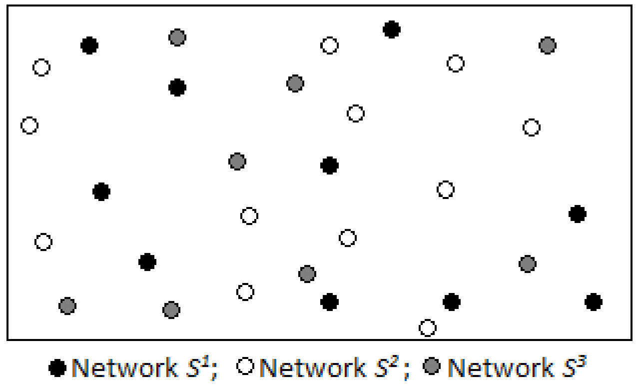

Figure 2.1 Three Collaborating Sensor Networks

tamper with the sensor in an attempt to gain control or learn information from the sensor.

The allied adversary can launch all of the same attacks as the external adversary; however, the effectiveness of these attacks is much different due to level of access provided to this adversary. First, the allied adversary already have access to some information in the network such as types of sensors available, sensor IDs, sensor locations, etc. Second, because information is shared between sensors in the network, the allied adversary has an established manner of communication with the sensors of the operator's network that includes encryption keys necessary for preventing an external adversary from reading the sensor networks' communications. These two factors allow for the allied adversary to have a more effective attack for some attacks such as eavesdropping, masquerade, and reflection attacks.

Since the goal is to securely authenticate sensors in the presence of an allied adversary and due to the elevated nature of the attacks that can be launched by an allied adversary, only the allied adversary will be considered during security analysis.

2.5. PHYSICALLY UNCLONABLE FUNCTIONS

Our proposed solution to the authentication problem proposes leveraging Physically Unclonable Functions. We assume each sensor in a network is provisioned with its own Physical Unclonable Function (PUF). A PUF is an innovative circuit primitive that provides a mechanism to extract secrets leveraging from physical randomness in hardware fabrication of integrated circuits (ICs) [11, 12, 13, 5, 14, 15]. More specifically, a PUF is a hardware primitive who behavior is determined by the physical structure of the hardware itself and its construction. The randomness of fabrication during circuit constructions makes no two circuits exactly the same. While a particular circuit exhibits repeatable behavior, predicting its performance before hand is not possible, and cloning of the circuit is highly impractical. Typically, PUFs are used in a challenge-response mechanism, wherein given a Challenge *C*, the PUF for a particular device will respond with a Response *R*. While *R* is repeatable for the same *C*, guessing or cloning the circuit to derive *R* is not possible hence providing a straightforward mechanism for hardware based authentication.

A number of properties of ICs today lend themselves to creating PUFs. An Optical PUF can be generated as a result of speckle patterns (intensity patterns produced by the mutual interference of a set of wave fronts) emanated when a laser beam shines on an optical material [16]. These patterns are random, unique, and unclonable, hence realizing an optical physically unclonable, hence realizing an optical PUF. Another type of PUF is called a Coating based PUF, where above a normal IC, a network of metal wires is laid out in a comb shape. The space between and above the comb structure is filled with an opaque material and randomly doped with dielectric particles. Because of the random placement, size, and dielectric strength of the particles, the capacitance between each couple of metal wires will be random up to a certain extent. A number of PUFs exploiting other physical properties that exhibit randomness during circuit fabrications have been designed exploiting inherent randomness during circuit fabrications. These include delay based PUF exploits random variations in delays of wires and gates on silicon [11, 12], oscillator frequencies [11, 13, 5], voltage values during power-up of SRAM (Static Random Access Memory) [14, 15].

Table 2.1 Properties of Physically Unclonable Functions

PUF Type	No. of Gates	No. of Bits	Response Time	Energy Per Response
Optical PUF [11, 12]		10^5	1ms	
Delay based Arbiter [11, 12]	450	2^{64}	5ns	0.239pJ
Ring Frequency Oscillator [11, 13, 5]	1159	496	1650ns	244.2pJ
SRAM Voltage based PUF [14, 15]	256	2^{50}	11ns	

With advances in hardware miniaturization, PUFs are becoming increasingly practical, with minimal overhead in space and energy expenditure. For instance, it is estimated that implementing a delay circuit requires about 6 to 8 gates for each input bit, and oscillating counter circuit that measures delay requires about 33 gates. Therefore, a 64-bit input delay PUF requires only about 545 gates [5]. A typical coating PUF has been implemented in [17] with just 1000 gates, and the optical PUF implemented in [16] can yield upto 10^6 challenge-response pairs with a delay of around 1 ms per authentication. The use of 256 SRAM blocks has been shown to yield 100 bits of true randomness each time the memory is powered up [15]. Note that the reliance on PUFS on subtle inherent physical variations during fabrication means that they are inherently sensitive to physical tampering [18, 19, 13, 20], and can be easily detected with incorrectly received responses after a circuit is tampered. Table 2 summarizes some PUF implementations and their properties.

3. OUR BASIC 3-WAY HANDSHAKING PROTOCOL

3.1. PROTCOL INTRODUCTION

We now present our basic 3-way handshaking protocol for authentication in collaborating sensor networks. We first present the description of the protocol, followed by an analysis on the security performance against attacks. We also assume that the total number of active sensors in Network *S* is m.

3.2. PROTOCOL DESCRIPTION

Protocol 1 presents our basic 3-way handshaking protocol. The protocol is executed each time (or round) when the Operator O^i intends to authenticate sensors belonging to network S^i. Consider an arbitrary Round *r*. Operator O^i will broadcast a query consisting of a Challenge Vector for that round:

$$\underrightarrow{\begin{bmatrix} N_{(r,1)}{}^{i} \parallel Y_1^i \oplus C_{(r,1)}{}^{i} \parallel Z_1{}^{i} \oplus A_{(r,1)}{}^{i} \\ N_{(r,2)}{}^{i} \parallel Y_2^i \oplus C_{(r,2)}{}^{i} \parallel Z_2{}^{i} \oplus A_{(r,2)}{}^{i} \\ \ldots \\ N_{(r,m)}{}^{i} \parallel Y_m{}^{i} \oplus C_{(r,m)}{}^{i} \parallel Z_m{}^{i} \oplus A_{(r,m)}{}^{i} \end{bmatrix}}$$

where $N_{(r,k)}{}^{i}$ is a nonce shared between Operator O^i and Sensor $s_k{}^i$. Once a sensor $s_k{}^i$ verifies that the nonce received is expected, it proceeds with the following steps. Otherwise, the message is discarded. Note that the nonce for the first round is pre-stored on sensor $s_k{}^i$. This completes the first part of the handshaking protocol. Using its secret keys pre-distributed keys $Y_k{}^i$ and $Z_k{}^i$, sensor $s_k{}^i$ extracts two challenges $C_{(r,k)}{}^{i}$ and $A_{(r,k)}{}^{i}$. Here $C_{(r,k)}{}^{i}$ denotes the challenge issued by the operator whose response from $s_k{}^i$ will then be used to authenticate it, while $A_{(r,k)}{}^{i}$ denotes the subsequent challenge whose response from O^i will enable sensor $s_k{}^i$ verify that its response was indeed received by O^i correctly. Once a sensor $s_k{}^i$ extracts $C_{(r,k)}{}^{i}$, it will compute a Response $P_{(r,k)}{}^{i}$ which is the output of the sensor's physically unclonable function, i.e., $P_{(r,k)}{}^{i} = PUF_k{}^i(C_{(r,k)}{}^{i})$. This response along with the sensor ID is then routed back to Operator O^i as $[s_k{}^i \parallel P_{(r,k)}{}^{i} \oplus Y_k{}^i]$. This completes the second part of the handshaking protocol.

Once Operator O^i receives responses (after a tolerable delay), it will verify if the received $P_{(r,k)}{}^i$ is the expected one for Sensor $s_k{}^i$. For every sensor whose response was correctly authenticated, the operator will derive $Q_{(r,k)}{}^i = PUF_k{}^i(A_{(r,k)}{}^i)$. For any sensor $s_j{}^i$ whose response cannot be verified as correct, $Q_{(r,j)}{}^i$ is set to a random bit string. This prevents any attackers targeting the unverified node to learn any information about the protocol by the absence of a message or data value. Operator O^i will broadcast this response to all sensors in the network in order to convince sensors receipt of their responses, along with the nonce for the next round. The message transmitted is

$$\begin{bmatrix} Q_{(r,1)}{}^i \oplus Y_1{}^i \,\|\, Q_{(r,1)}{}^i \oplus N_{(r+1,1)} \oplus Z_1{}^i \\ Q_{(r,2)}{}^i \oplus Y_2{}^i \,\|\, Q_{(r,2)}{}^i \oplus N_{(r+1,2)}{}^i \oplus Z_2{}^i \\ \ldots \\ Q_{(r,m)}{}^i \oplus Y_m{}^i \,\|\, Q_{(r,m)}{}^i \oplus N_{(r+1,m)}{}^i \oplus Z_m{}^i \end{bmatrix}$$

Each sensor $s_k{}^i$ can now verify if its message was indeed received correctly by verifying the correctness of $Q_{(r,k)}{}^i$, based on the challenge $A_{(r,k)}{}^i$ that it already possesses. Each sensor will also be able to successfully extract the expected Nonce $N_{(r+1,k)}{}^i$ for the next round $r + 1$. If $Q_{(r,k)}{}^i$ for sensor $s_k{}^i$ is not the expected value, this means that O^i did not receive the sensor's response due to possible packet drop or corruption enroute. Hence Sensor $s_k{}^i$ will send its original $P_{(r,k)}{}^i$ via multiple routing paths to the operator expecting an acknowledgement. If an acknowledgement from O^i still does not arrive, the sensor can practically consider itself in-active due to a broken communication link with the operator. This completes the 3-way handshaking protocol.

3.3. ANALYSIS OF THE PROPOSED PROTOCOL

In this section, we present a security analysis of Protocol 1 against attacks discussed in Section 2.3.

3.3.1. Eavesdropping Attacks. The attacker can eavesdrop on any communication in the network. However, the adversary will not be able to infer any information that could compromise the authentication process. By observing the PUF responses $P_{(r,k)}^i$ of sensor s_k^i in Round r, the adversary will not be able to infer anything useful about the current or subsequent communication since PUF responses cannot be predicted in advance or cloned. Note that it may be possible that the adversary may infer the number of sensors belonging to a network by observing the number of responses. Such an attack can be easily thwarted if Operator O introduces dummy entries in its queries, and if sensors send dummy message during query responses. Dummy queries will not be processed, while dummy responses from sensors will be identified by the operator and discarded. The downside though may be increased overhead during the messages forwarding.

Also, an eavesdropping adversary may capture messages from the operator. However, since messages are encrypted using secret keys Y_k^i and Z_k^i for sensor s_k^i, the adversary will not be able to infer Challenges $C_{(r,k)}^i$ or $A_{(r,k)}^i$ for Round r (Step 1). Similarly, the adversary can eavesdrop on the response message of the operator for Round r (Step 19). The adversary could then attempt to discover information by observing the plain text nonce in the next round. First, the adversary will not be able to infer $N_{(r+1,k)}^i$ for Sensor s_k^i from any passive observations in Round r due to encryption. Also, by performing operations $Q_{(r,k)}^i \oplus Y_k^i \oplus Q_{(r,k)}^i \oplus N_{(r+1,k)}^i \oplus Z_k^i \oplus N_{(r+1,k)}^i$, the adversary will only be able to infer $Y_k^i \oplus Z_k^i$, which itself yields no useful information about the keys stored.

3.3.2. Masquerading Attacks. Attackers may impersonate sensors in the network during querying. However, a masquerading sensor will not be to generate the correct PUF response. Such messages will be identified as fake by the operator and discarded automatically. Note that since PUFs cannot be cloned due to their inherent randomness during fabrication, circuit cloning attacks are infeasible.

Protocol 1 Basic 3-way Handshaking Protocol in Round r

Operator O^i sends $\left[\begin{array}{l} N_{(r,1)}{}^i \,\|\, Y_1^i \oplus C_{(r,1)}{}^i \,\|\, Z_1{}^i \oplus A_{(r,1)}{}^i \\ N_{(r,2)}{}^i \,\|\, Y_2{}^i \oplus C_{(r,2)}{}^i \,\|\, Z_2{}^i \oplus A_{(r,2)}{}^i \\ \ldots \\ N_{(r,m)}{}^i \,\|\, Y_m{}^i \oplus C_{(r,m)}{}^i \,\|\, Z_m{}^i \oplus A_{(r,m)}{}^i \end{array}\right] \xrightarrow{\hspace{2em}}$ to sensors

End 1-way handshake

Each Sensor s_k^i executes the following steps

IF $N_{(r,k)}{}^i$ is as expected

    Extract $C_{(r,k)}{}^i$ and $A_{(r,k)}{}^i$

    Compute $P_{(r,k)}{}^i = PUF_k^i(C_{(r,k)}{}^i)$

    s_k^i sends $[s_k^i \,\|\, P_{(r,k)}{}^i]$ to O^i

ELSE Reject Request

END IF

End 2-way handshake

Operator O^i executes the following steps for each Sensor s_k^i

IF received $P_{(r,k)}{}^i$ matches expected response

    Authenticated Sensor s_k^i as Active

    Compute $Q_{(r,k)}{}^i = PUF_k^i(A_{(r,k)}{}^i)$

ELSE

    Consider Sensor s_k^i as Inactive

    Set $Q_{(r,k)}{}^i$ = Random bit string

END IF

Operator O^i $\left[\begin{array}{l} Q_{(r,1)}{}^i \oplus Y_1^i \,\|\, Q_{(r,1)}{}^i \oplus N_{(r+1,1)}{}^i \oplus Z_1^i \\ Q_{(r,2)}{}^i \oplus Y_2{}^i \,\|\, Q_{(r,2)}{}^i \oplus N_{(r+1,2)}{}^i \oplus Z_2{}^i \\ \ldots \\ Q_{(r,m)}{}^i \oplus Y_1^i \,\|\, Q_{(r,m)}{}^i \oplus N_{(r+1,m)}{}^i \oplus Z_m{}^i \end{array}\right] \xrightarrow{\hspace{2em}}$

Each Sensor s_k^i executes the following steps

IF $Q_{(r,k)}{}^i = PUF_k^i(A_{(r,k)}{}^i)$

    Extract Nonce $N_{(r+1,1)}{}^i$ for Round $r + 1$

ELSE Send $P_{(r,k)}{}^i$ to O^i via multiple routing paths

END IF

End 3-way handshake

3.3.3. Reflection Attacks. Reflection attacks are not a threat to the authentication process in Protocol 1, since the challenges and response of sensors and the operator are different. Sensors will respond with the PUF value only upon correctly verifying the nonce from the operator which are not exposed to the adversary. Similarly the adversary will never gain knowledge of, or generate the PUF value that can be used for a subsequent authentication process. Even if the adversary captures the PUF response for a challenge, the same challenge is very unlikely to be used again for sufficiently long challenge bit sequences. As pointed earlier in Table 2, up to 2^{64} challenge-responses are feasible with PUFs today. Hence reflection attacks are addressed in Protocol 1.

3.3.4. Packet Drop/ Packet Corruption Attacks. In a network of collaborating sensor networks, any of the messages sent by the operator or the sensor may travel through sensors belonging to other networks. In this scenario the message may be dropped, potentially disrupting the authentication process. This can be easily detected because, at each phase of the protocol, a message is expected by either the operator or the sensor. If that message does not arrive, the sensor and/or operator can resend its previous message with or without modifications to the routing path until a set number of retry attempts is met, at which point the sensor can be considered inactive due to the inability to successfully communicate with the sensor. The same responses and consequences will also be used if a packet corruption attack is used instead. In either case, this does not compromise the correctness of Protocol 1. These attacks are similar to a denial of service attack except that instead targeting the sensor, the routing path is targeted.

3.3.5. Replay and Selective Forwarding Attacks. Replay and Selective Forwarding attacks can be launched by other malicious sensors that have eavesdropped on previous packets. However, since the PUF responses are unique to every challenge, there is no incentive for adversaries to launch such attacks. An adversary that attempts to launch replay or selective forwarding attacks will be ignored by the sensors, since the expected nonce will never match. The energy consumed for comparing a sequence of bits is very minimal in sensors. Furthermore, if repeated replay and selective fowarding attacks are launched, it is easy for sensors to detect the presence of an adversary and notify the operator who can then take other corrective actions.

3.3.6. Denial of Service Attacks. In the event that attackers are able to jam one or more sensors in the network, their responses will not be able to reach the operator. As long as a jamming attack continues, the sensor being jammed is practically useless, and hence it will not be considered as an active sensor by the operator.

3.3.7. Physical Attacks. As pointed earlier in the section, PUFs provide an inherent resilience against physical tampering [18, 19, 13, 20]. When adversaries physically tamper with sensors, the physical characteristics of the circuit will be altered, and the PUF responses to challenges will also be altered. Upon receiving incorrect PUF responses to challenges, the operator will subsequently identify a physically tampered sensor as inactive.

As shown, our protocol is highly resilient against a variety of attacks. While some of the attacks can disrupt and block communications with a given sensor, the allied adversary is still unable to break the authentication protocol outlined above. This also holds true for an external adversary since their attacks will not have the level of access available to the allied adversary.

4. OUR AGGREGATED 3-WAY HANDSHAKING PROTOCOL

4.1. PROTOCOL INTRODUCTION

Our basic 3-way handshaking protocol presented above is robust against a number of attacks compromising authentication in collaborating sensor networks. However, the major limitation of Protocol 1 is that each sensor individually forwards its response to the operator. This will introduce significant communication overhead in large scale networks, which our proposed 3-way Aggregated Protocol described below alleviates without compromising security performance. We also assume that the total number of active sensors in Network S^i is m.

4.2. PROTOCOL DESCRIPTION

Protocol 2 presents our aggregated 3-way handshaking protocol. This protocol considers a sensor network clustered into a certain number of clusters. Each sensor belongs to one cluster with a cluster-head. The operator is assumed to know which sensor belongs to which cluster, which could be known just after deployment as sensors generate clusters among themselves using techniques in [21, 22]. Protocol 2 is executed each time (or round) when the Operator O^i intends to authenticate sensors belonging to network S^i. Consider an arbitrary Round r. The operator will broadcast a query vector containing the nonce and encrypted challenge vectors for each sensor. This completes the first part of the handshaking protocol.

After computing the PUF response $P_{(r,k)}{}^i = PUF_k{}^i(C_{(r,k)}{}^i)$, each sensor will forward its response to its cluster-head. Consider Cluster j for illustration. The cluster-head will aggregate all responses in its cluster using the XOR function to compute $G_{(r,j)}{}^i$ for Cluster j. It will then broadcast $G_{(r,j)}{}^i$ and responding sensor IDs to its upstream cluster-head and all sensors in its cluster. The upstream cluster-head will once again perform aggregation and this process continues towards the operator. Each sensor in Cluster j will store $G_{(r,j)}{}^i$ for subsequent verification in the event of a packet corruption. This completes the second part of the handshaking protocol.

Protocol 2 Aggregated 3-way Handshaking Protocol in Round r

1: A Sensor Network Clustered into J Clusters

2: Operator $O^i \xrightarrow{\begin{bmatrix} N_{(r,1)}{}^i \,\|\, Y_1^i \oplus C_{(r,1)}{}^i \,\|\, Z_1{}^i \oplus A_{(r,1)}{}^i \\ N_{(r,1)}{}^i \,\|\, Y_1^i \oplus C_{(r,1)}{}^i \,\|\, Z_1{}^i \oplus A_{(r,1)}{}^i \\ \\ N_{(r,m)}{}^i \,\|\, Y_m{}^i \oplus C_{(r,m)}{}^i \,\|\, Z_m{}^i \oplus A_{(r,m)}{}^i \end{bmatrix}} *$

3: **End 1-way handshake**
4: Each Sensor s_k^i executes the following steps
5: IF $N_{(r,1)}{}^i$ is as expected
6: IF Sensor s_k^i is a NOT a Cluster-Head
7: Extract $C_{(r,k)}{}^i$ and $A_{(r,k)}{}^i$
8: Compute $P_{(r,k)}{}^i = PUF_k^i(C_{(r,k)}{}^i)$
9: s_k^i [s_k^i || $P_{(r,k)}{}^i$] Cluster-Head
10: ELSE IF Sensor s_k^i is a Cluster-Head of Cluster j
11: Extract $C_{(r,k)}{}^i$ and $A_{(r,k)}{}^i$
12: Compute $P_{(r,k)}{}^i = PUF_k^i(C_{(r,k)}{}^i)$
13: Compute $G_{(r,j)}{}^i = P_{(r,k)}{}^i \oplus P_{(r,j)}{}^i$ $\forall$ responding sensors s_j^i in Cluster j
14: Forward $G_{(r,j)}{}^i$, Sensor IDs to Peer Sensors and Upstream Cluster-Head
15: END IF
16: ELSE Reject Request
17: END IF
18: **End 2-way handshake**
19: Operator O^i executes the following steps for each Sensor s_k^i
20: Compute $\forall$ responding sensors s_k^i, $\oplus P_{(r,k)}{}^i$
21: IF Response matches Expected Response
22: Authenticated All Sensor IDs received as Active
23: Compute $Q_{(r,k)}{}^i = PUF_k^i(A_{(r,k)}{}^i)$
24: ELSE % malicious behavior detected
25: Operator computes $\forall$ responding sensors s_k^i in Cluster j, $G_{(r,j)}{}^i = \oplus P_{(r,k)}{}^i$
26: Operator $O^i \xrightarrow{\begin{bmatrix} G_{(r,1)}{}^i \\ G_{(r,2)}{}^i \\ \ldots \\ G_{(r,m)}{}^i \end{bmatrix}} *$
27: FOR Each Cluster j in the Network
28: FOR Each sensor s_k^i in Cluster j
29: Report $P_{(r,k)}{}^i = PUF_k^i(C_{(r,k)}{}^i)$ to Operator
30: END FOR
31: END FOR
32: Operator can identify malicious sensors and consider them inactive
33: Operator sets $Q_{(r,k)}{}^i$ = Random bit string $\forall$ inactive sensors s_k^i

Continued on next page.

Protocol 2 Continued

    Operator sets $Q_{(r,k)}{}^{i}$ = Random bit string $\forall$ inactive sensors $s_k{}^i$
END IF
Operator O^i $\xrightarrow{\begin{bmatrix} Q_{(r,1)}{}^{i} \oplus Y_1^i \,\|\, Q_{(r,1)}{}^{i} \oplus N_{(r+1,1)}{}^{i} \oplus Z_1{}^{i} \\ Q_{(r,2)}{}^{i} \oplus Y_2{}^{i} \,\|\, Q_{(r,2)}{}^{i} \oplus N_{(r+1,2)}{}^{i} \oplus Z_2{}^{i} \\ \ldots \\ Q_{(r,m)}{}^{i} \oplus Y_1^i \,\|\, Q_{(r,m)}{}^{i} \oplus N_{(r+1,m)}{}^{i} \oplus Z_m{}^{i} \end{bmatrix}}$
Each Sensor $s_k{}^i$ executes the following steps
IF $Q_{(r,k)}{}^{i} = PUF_k{}^i(A_{(r,k)}{}^i)$
    Extract Nonce $N_{(r+1,\ 1)}{}^{i}$ for Round $r + 1$
ELSE Send $P_{(r,k)}{}^{i}$ to Operator via multiple routing paths
END IF
End 3-wav handshake

Upon receiving the aggregated response and all responding sensor ids, the operator will compute $\oplus P_{(r,k)}{}^{i}$ for responding sensors $s_k{}^i$. If this value matches the aggregated response received from the immediate downstream cluster-head(s), the authentication process is completed and all responding sensors are considered active. Otherwise, at least one or more sensors generated a malicious response or an intermediary malicious sensor processed a packet and corrupted it. In either case, the operator will broadcast the expected aggregated responses to each cluster. When a sensor receives $G_{(r,j)}{}^{i}$ from the operator for its cluster, it will compare the correctness of its own aggregated response that was forwarded to it by its own cluster-head. If the compared values match, then there was no malicious sensor in its cluster and the sensor ignores the message. If on the other hand, the compared values do not match, then the sensors in the cluster will send their individual responses to the operator and the operator can now detect the malicious response, since the sensor that was the source of the malicious behavior will not be able to generate the correct PUF response. Similarly, corruptions of packets by intermediate sensors can also be detected. After this step, all active sensors will be correctly authenticated and the operator will send an authentication response along with an encrypted form of the nonce to be used for the next round of authentication. This completes the 3-way handshaking protocol.

4.3. ANALYSIS OF THE PROPOSED PROTOCOL

Because of the similarities to Protocol 1, Protocol 2 and Protocol 1 share many of the same security properties. Their defense against eavesdropping, reflection, DOS, and physical attacks are the same in every respect except the time frame in which the attack can occur. This occurs because the aggregation process requires an additional amount of time that is missing from Protocol 1. Assuming that the networks are the same, Protocol 2 will usually take longer than Protocol 1.

4.3.1. Replay, Selective Forwarding, Packet Drop and Corruption Attacks. These attacks gain a slight advantage in Protocol 2 due to the increase in the number of message transmissions. Each of these attacks requires manipulating a target packet via duplication, denial, or corruption, and so any increase in the number of packets being sent in a protocol automatically increases the number of viable targets that these attacks can be initiated on. In Protocol 2, 2 new message exchanges are introduced: the request for specific PUF responses when a malicious node is detected and the subsequent reply. Both of these messages can be targeted by the adversary. Unfortunately, that is the only advantage that the adversary gains. The results of these attacks will be the same as in Protocol 1 due to the packets being resent along different routes in the case of selective forwarding and packet drop attacks and the requirement of a particular PUF response during this exchange in the case of packet corruption and replay attacks.

4.3.2. Masquerade Attacks. The advantage gained by a masquerade attack is the same as for the replay, selective forwarding, packet drop and corruption attacks as described above: because of the increase in message exchanges, more opportunities are available for the masquerading node to initiate an attack. But again, the attack is still thwarted by the requirement of specific PUF responses that a masquerading node cannot reproduce.

As shown, even with providing additional opportunities for an adversary to attack, Protocol 2 is still resilient to a variety of attacks. There is the vulnerability to having communications disrupted and blocked completely as mentioned in Protocol 1, but this still does not inherently break the authentication protocol.

4.4. DISCUSIONS OF IMPROVING SCALABILITY AND MALICIOUS SENSOR IDENTICATION

Currently, when a single, aggregated response is returned to the operator containing a malicious response, the entire cluster tree is queried to identify the exact sensors that have been compromised. This introduces considerable overhead into the network and also provides more opportunities for the adversary to compromise the authentication process by targeting the messages being transmitted. By modifying how the PUF responses are XOR'ed together, malicious sensors can be detected before steps 25 – 32 and / or reduce the number of potentially compromised nodes that must be checked. Instead of $\oplus$ all $P_{(r,k)}^{i}$, each cluster head $\oplus$ its $P_{(r,k)}^{i}$ with each received response and then forwards this group response to the upstream cluster head. This results in each leaf sensor being $\oplus$ with all its cluster heads in its tree branch. Additionally, each cluster head can be uniquely identified by its sensor groupings. This allows for the operator to identify where in the sensor network a malicious response was inserted, reducing the number of sensors that must be queried if a cluster head is compromised and providing immediate detection of compromised leaf sensors. The scalability of the network is also improved as this greatly reduces the communication overhead that would be incurred from querying entire cluster branches and improves detection as more leaf sensors and cluster heads are added. The only downside to this fix is that the base communication overhead in the aggregated PUF responses is increased by the number of leaf sensors.

5. CONCLUSIONS

5.1. CONCLUSION

In this thesis, we addressed the problem of authentication in collaborating sensor networks. Our protocols are based on Physically Unclonable Functions, an innovative circuit primitive that provides a mechanism to extract secrets leveraging from physical randomness in hardware fabrication of integrated circuits (ICs). Our protocols are light-weight, efficient, correct and highly resilient to a variety of attacks. Addressing other security, privacy problems in collaborating sensor networks is part of future work.

BIBLIOGRAPHY

[1] M. Hassan, B. Song, and E. Huh, "A framework of sensor-cloud integration opportunities and challenges," in *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication.* ACM, 2009, pp. 618-626.

[2] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure-physical-sensor management with virtualized sensors on cloud computing," in *Network--Based Information Systems (NBiS), 2010 13th International Conference on.* IEEE, 2010, pp. 1-8.

[3] Y. Xu, S. Helal, M. Thai, and M. Scmalz, "Optimizing push/pull envelopes for energy-efficient cloud-sensor systems," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems.* ACM, 2011, pp. 17-26.

[4] S. Nath, J. Liu, and F. Zhao, "Sensormap for wide-area sensor webs," *Computer*, vol. 40, no. 7, pp. 90-93, 2007.

[5] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight secure search protocols for low-cost rfid systems," in *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Confferent on.* IEEE, 2009, pp. 40-48.

[6] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Network and Distributed System Security Symposium (NDSS),* San Diego, February 2003.

[7] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS),* October 2003.

[8] W. Gu, N. Dutta, S. Chellappan, and X. Bai, "Providing end-to-end secure communications in wireless sensor networks," *IEEE Transactions on Network and Service Management (TNSM), to appear.*

[9] D. Malan, M. Welsh, and M. Smith, "Implementing public-key infrastructure for sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 4, p. 22, 2008.

[10] L. Yao, Z. Yu, T. Zhang, and F. Gao, "Dynamic window based multihop authentication for wsn," in *Proceedings of the 17th ACM conference on Computer and communications security.* ACM, 2010, pp. 744-746.

[11] S. Srivathsa, "Secure and energy efficient physical unclonable functions," Ph.D. dissertation, University of Massachusetts Amherst, 2012.

[12] I. Verbauwhede and R. Maes, "Physically unclonable functions: manufacturing variability an unclonable device identifier," in *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI.* ACM, 2011, pp.455-460.

[13] H. Handschuh, G. Schrijen, and P. Tuyls, "Hardware intrinsic security from physically unclonable functions," *Towards Hardware-Intrinsic Security,* pp. 39-53, 2010.

[14] R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for sram pufs," *Cryptographic Hardware and Embedded Systems-CHES 2009,* pp. 332-347, 2009.

[15] R. Colopy, "Sram characteristics as physical unclonable functions," Ph.D. dissertation, Worcester Polytechnic Institute, 2009.

[16] P. Tuyls, B. Škorić, S. Stallinga, A. Akkermans, and W. Ophey, "Information-theoretic security analysis of physical uncloneable functions," *Financial Cryptography and Data Security,* pp. 578-578, 2005.

[17] R. Maes and I Verbauwhede, "A discussion on the properties of physically unclonable functions," *status: published,* 2010.

[18] F. Armknecht, R. Maes, A. Sadeghi, B. Sunar, and P. Tuyls, "Puf-prfs: a new tamper-resilient cryptographic primitive," in *Advances in Cryptology-EUROCRYPT,* vol. 2009, 2009, pp. 96-102.

[19] E. Ozturk, G. Hammouri, and B. Sunar, "Physical unclonable function with tristate buffers," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on.* IEEE, 2008, pp. 3194-3197.

[20] G. Hammouri, E. Öztürk, and B. Sunar, " A tamper-proof and lightweight authentication scheme," *Pervasive and Mobile Computing,* vol. 4, no. 6, pp. 807-818, 2008.

[21] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *Mobile Computing, IEEE Transactions on,* vol. 3, no. 4, pp. 366-379, 2004.

[22] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies,* vol. 3. IEEE, 2003, pp. 1713-1723.

[23] M. Younis, K. Ghmman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 17, no. 8. Pp. 865-882, 2006.

[24] A. Ferreira, M. Vilaça, L. Oliveira, E. Habib, H. Wong, and A. Loureiro, "On the security of cluster-based communication protocols for wireless sensor networks," *Networking-ICN 2005*, pp. 449-458, 2005.

[25] M. Ba, I. Niang, B. Gueye, and T. Noel, "A deterministic key management scheme for securing cluster-based sensors networks," in *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on.* IEEE, 2010, pp. 422-427.

[26] V. Anh and A. Moffat, "Inverted index compression using word-aligned binary codes," *Information Retrieval,* vol. 8, no. 1, pp. 151-166, 2005.

[27] A. Moffat and L. Stuiver, "Binary interpolative coding for effective index compression," *Information Retrieval*, vol. 3, no. 1, pp. 25-47, 2000.

[28] —, "Exploiting clustering in inverted file compression," in *Data Compression Conference, 1996. DCC'96. Proceedings.* IEEE, 1996, pp. 82-91.

[29] S. Chellappan, V. Paruchuri, D. McDonald, and A. Durresi, "Localizing sensor networks in unfriendly environments," in *IEEE Military Communications Conference (MILCOM),* San Diego, November 2008.

[30] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the ACM Conference on Computer and Communication Security (CCS)*, November 2002, pp. 41-47.

[31] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of IEEE Symposium on Research in Security and Privacy,* May 2003.

VITA

Jake Uriah Bielefeldt has earned a B.S. degree in Computer Science in May of 2013 from Missouri University of Science and Technology, and will earn a M.S. degree in Computer Science from Missouri University of Science and Technology in May of 2015.