Enhancements of Security in Wireless Sensor Networks

Brian D. Caravantes
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
carav008@morris.umn.edu

ABSTRACT

Wireless sensor networks (WSNs) are used in everyday settings from military applications to health monitoring. WSNs are networks that are made up of tiny sensors that act as miniature computers used to collect and gather data in almost any environment. However, there are concerns about the security in WSNs because of the resource restrictions. This paper will discuss two different methods that address security concerns in WSNs.

Keywords

Wireless Sensor Networks, Body Area Networks, Authentication, Cryptography, Security

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of sensor nodes that have computation and communication capabilities [10]. These nodes gather and send the data to a base station (gateway) node which then can use the data locally or allow for the data to be accessed by some user in real-time. The base station simply acts as a gateway for communication between the user and some node.

WSNs are used today in diverse applications including environmental applications, health, and home monitoring as well as military operations [9]. A real-life example of a typical WSN is in place at the University of Minnesota Morris (UMM). UMM's on campus living residence, the Green Prairie Community (GP), is an environmentally friendly living residence that uses a WSN to collect electricity usage data about each room. A base station is located in GP's basement. If an authorized user of UMM wanted to access data about a certain rooms electricity usage, they would have authenticate through the base station before being able to access real-time data about that room. Figure 1 shows a typical WSN setup.

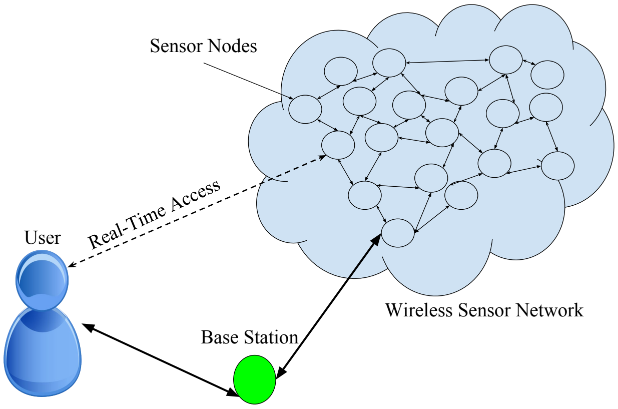

Figure 1: A typical WSN setup: Note that the base station is used for authentication to grant real-time access to data.

Since WSNs consist of a large network of interconnected components, security is a major issue when it comes to these types of networks [9]. WSNs face *resource restrictions* that include storage capacity, processing power and battery power that make it difficult to ensure that security requirements are met. These security requirements are broken down into categories of data integrity, confidentiality, and availability. Data confidentiality deals with ensuring transmitted data in the network is limited to intended users only. [9] Data integrity aims assure that the data sent isn't changed during transmission to original destination. Lastly, availability ensures that the network is able to provide services at any time for an authorized user. Security in WSNs, as it stands now, fails to meet all three of these requirements, but the proposed protocols in this paper aim to satisfy these requirements.

The rest of the paper is organized as follows. In Section 2 we describe background information needed to understand the proposed methods. We continue in Section 3 by looking at how the method works as well as analyzing its security enhancements in WSNs. Then in Section 4 we look at the second method of enhancing WSN security and conclude our findings in Section 5.

2. BACKGROUND

In this section we will cover background information needed to understand the proposed protocols for enhancing WSN security. One must understand WSN deployment settings, the attacks that WSNs most commonly face, as well as cryptographic elements used in each proposed scheme.

UMM CSci Senior Seminar Conference, April 2018 Morris, MN.

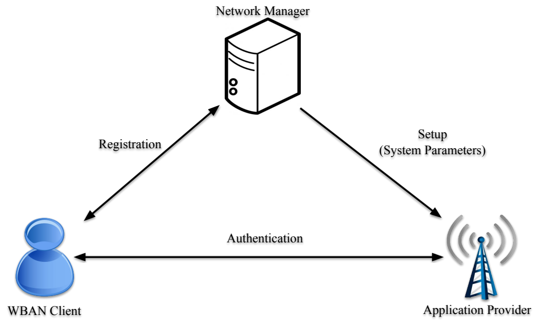

Figure 2: A simplified diagram of how WBANs work [3].

2.1 Deployment settings for WSNs

Deployment settings play a crucial role in WSNs because where WSNs are used depend on what is needed by a user [7]. I have chosen to focus on two types of deployment settings, health monitoring and military operations, each of which will be built upon later in this paper.

2.1.1 Health monitoring

Wireless body area networks (WBANs) are considered a subset of WSNs, although WBANs face additional challenges related to human body environments and interactions [4]. WBANs consist of a typical WSN but the nodes are placed in or around the human body and used for various applications. WBANs collect real-time bio-medical data from the human body, which are transmitted to a remote medical server through mobile devices such as a smart phone, where the medical staff can use them for patient evaluation [3].

To explain the process, some client or user C will access the WBAN while a Network Manager NM, a trusted third party, will provide the system parameters. The system parameters are just credentials needed for C to recognize NM. The Application Provider AP is usually a remote system such as a server or medical system at a hospital, responsible for providing medical services. Figure 2 displays how the a WBAN interaction might work.

In health monitoring, if WBANs fail to provide secure services very sensitive and important data can be captured by an unauthorized user. Additional concern also arises when bad actors try to control the network. Privacy should be provided throughout WBAN communication to ensure highly confidential bio-medical data isn't leaked or captured by anyone who shouldn't have it. This is why enhancing security in WBANs is essential.

2.1.2 Military deployment

In military settings there is always the chance of being attacked by an enemy. This makes WSNs useful because of their capability of real time transmission [6]. Most military settings are infrastructure based, meaning that the WSNs are placed in or around organizational structures. This can include military operations like border fence monitoring, critical installation monitoring and mine field protection.

WSNs deployed in military settings have the usual security requirements (integrity, confidentiality and availability) as mentioned in Section 1. Security is essential for military operations; if WSNs fail to provide secure communications, consequences can include problems in sighting enemies across borders or not detecting nearby minefields. Therefore providing a protocol for enhancing security in military operations is needed.

2.2 Vulnerabilities and Attacks

Regardless of what type of deployment setting is chosen, attacks on networks still happen. WSNs are no exception. In this section, we will talk about common attacks networks face and how they affect WSNs. Network attacks may have different goals. Some try to access unauthorized information, some try to get control of the nodes in a network, and some just try to disrupt services and communication.

The goal of a Replay Attack is to access unauthorized information by recording parts of a session and resending them later [8]. The hope is that the recipient treats the replayed messages as new messages. In terms of WSNs this can be dangerous. Suppose a user tries to access sensitive physiological information from some medical service via a WBAN. If some adversary captures an old message, this will allow them to use it and request a response containing potentially sensitive information.

An Impersonation Attack occurs when adversary attempts to impersonate one of the other parties in a given protocol [8]. The goal is to access unauthorized information by acting as a certain party. Suppose an adversary obtains user credentials to impersonate a certain user. It would be quite easy for them to request or send falsified data over a WSN.

Node tampering or capturing is another potential problem in WSNs. The goal is to get control of an individual or cluster of nodes. Once some adversary has control of any number of nodes, it isn't difficult for them to add, remove, or alter information about a node.

2.3 Cryptography in WSNs

Cryptography is an important component of WSN security. In this section, we will explore various aspects of cryptography needed to understand the protocols discussed in this paper.

Cryptography is used heavily in WSNs to hide the meaning of a message sent between any combination of user, base station, and nodes. Cryptography is classified into private key (symmetric key) and public key (asymmetric key). [7] A majority of the research community working in WSNs state that symmetric key is favored because public key is too resource intensive [7].

We will lightly discuss public key cryptography for the understanding of material addressed in Section 3.

2.3.1 Private Key Cryptography

Private key cryptography involves two parties that are trying to communicate where each has an encryption and decryption method for which they share a secret key. [5] A key is just some form of data that is used for encryption and decryption. Encryption and decryption are functions that take in the key and a message as a parameter and return some specified output. Private key cryptography is best explained by example. Imagine a party X trying to send m with the message “hello” to party Y. The encryption algorithm in our example will be that each letter in the alphabet is now replaced by the third letter down the alphabet. This means ‘a’ is equal to ‘d’, ‘b’ is equal to ‘e’ and so on. One must note that this algorithm is a very simple example and

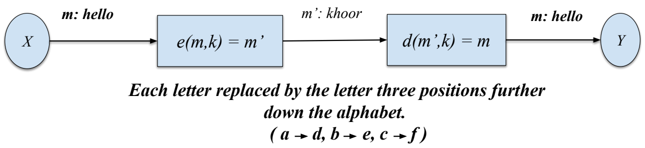
m: hello
X
e(m,k) = m'
m': khoor
d(m',k) = m
m: hello
Y
Each letter replaced by the letter three positions further down the alphabet.
(a → d, b → e, c → f)

Figure 3: Simple cryptographic process where *e(m,k)* and *d(m,k)* are the encryption and decryption functions.

not at all secure. X will encrypt the message m with the key k where k is the number of letters by which we shift, meaning 3.

X will then send the encrypted message m' to Y for decryption. Figure 3 shows this simple process in its entirety.

2.3.2 Public Key Cryptography

Public Key cryptography, also known as asymmetric cryptography, is a type of cryptography that, unlike private key cryptography uses two keys, a public and private (or secret key). [5] The public key can be published and known to everyone; the private key is only known to the key pair owner. Only someone with the the public/private key pair can decrypt received messages. The public and private key are related by mathematical properties, but it's impossible to find out the private key from the public key.

2.3.3 Other Cryptographic Elements

A hash function h transforms input data of any length into output bit string of specific fixed length. [5] The hash value is the returned value of the hash function; it is the unique representation or "finger print" of a message. Thus given a hash output z it must be computationally infeasible to find an input message x such that $z = h(x)$. The hash value is used for verification of some type of information. Imagine a party wants to verify information on some physical document. But the document contains sensitive information thus sending it over a network comes with risk. A way to verify is for the sender of the document to hash the contents of the document and send the resulting hash value. If the receiver hashes their document and it outputs the same hash value, then they will know that the content of the document has not been modified.

XOR or exclusive-or operator is another cryptographic aspect used heavily in cryptography. It is denoted by $\oplus$ and it is useful for encrypting a string of bits. XOR simply means one or the other but NOT both. Suppose we have *1101*$\oplus$*1010*, going through elements of each of the bit strings and comparing each one, the resulting string would be *0111*. The process returns 1 if we have a 1 and a 0 but not both the same. An important property of XOR used later in this paper with respect to encryption is that it's reversible, i.e., where C = A$\oplus$B, then you can get back A using A = C$\oplus$B.

Lastly concatenation is a cryptographic aspect that is used frequently in cryptography. It is denoted by || and just stands for the joining of multiple given strings. This means that if have 11||00, it will yield the result 1100.

3. ANONYMOUS AUTHENTICATION PROTOCOLS FOR WSNs

In this section we will discuss a 4 phase anonymous authentication (AA) protocol proposed by Gope and Hwang. [2] We will then analyze the enhancements in security it provides as well as discuss a similar AA protocol applied specifically for WBANs.

3.1 Proposed 4 Phase AA Protocol

The 4 phase AA Protocol proposed by Gope and Hwang [2] builds upon previous two-factor authentication schemes to enhance security in WSNs. Two-Factor Authentication is an authentication method that adds an extra layer of security by requiring not only a username and password for a user to be authenticated, but also for some other information that only the user has such as some physical token [1]. In the 4 Phase AA protocol the physical token is a smart card. A smart card is just some container, usually a "card" per se, that will store sensitive key material [5]. The 4 phases are somewhat independent of each other but each part improves security in its own respective manner.

3.1.1 Phase 1: Registration Phase

The registration phase is the first of the four phase process in the proposed AA scheme. This phase is necessary for letting some user "register" with a base station. [2] The scheme starts with some user U providing their unique identity credentials ID_U to the base station node (B) through some secure channel. B will play a crucial part in this entire process by producing various types of keys including private and session keys. The components inside h were specifically chosen because only U and B will have these, keeping the shared key secret. Also B as the base station, is some server with a database in remote location; ID_B is thus the identity of the base station that is found on the hardware of B. B will produce a private key K_{ub} which is shared key between U and B. $K_{ub} = h(ID_U||n) \oplus ID_B$ where n is some random 128-bit number produced by U.

Afterwards, B produces a transaction sequence number TS_{ub}, which is just a 64-bit number that will be passed around and is computed for each request m the user makes to B. [2] TS_{ub} is incremented by one after said request, stored in B's database and an encoded version is sent to U. The concept of TS_{ub} is used mainly to speed up the authentication process as well as prevent a replay attempt from any adversary. This transaction number will be used more in Phase 2. Since this is a type of two factor authentication, the base station node B will set up U's smart card SC with the parameters including the shared key K_{ub}, TS_{ub} and h. Subsequently B uses its secret key ω to store encoded versions of the parameters, $\{K_{ub}, TS_{ub}, ID_U\}$, in its database for future use.

3.1.2 Phase 2: Anonymous Authentication and Key Exchange

The purpose of this phase is to authenticate the user to both the base station and the sensor node the user wants to communicate with [2]. This phase is split up into 4 steps. One must note the difference between Phase 1 and 2, where the user is first registering with B vs trying to communicate with some sensor node S in the network. Figure 4 provides a visual representation of the 4 steps in phase 2.

In each step, a request message M_i where i goes from 1

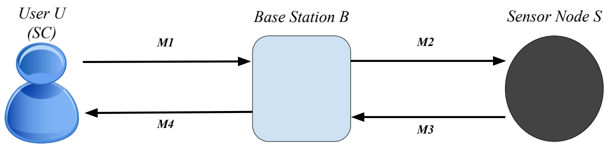

Figure 4: 4 Steps of Phase 2 (AA part)

to 4 (for each step of communication) provides an encryption specification. Before the user U first sends the request message M_1 to B, U will provide their identity credentials ID_U and password PSW_U to the smart card (SC) terminal where SC will compute the decoded shared key K_{ub} and the user function f_U through a one way hash function. Verification will occur if the SC's stored decoded user function is equal to the newly decoded f_U. Once verification happens the SC creates a one-time alias identity $AID_U = h(ID_U||K_{ub}||N_U||TS_{ub})$ where N_U is a number randomly generated by SC; AID_U will be used throughout the entire M_i process as a parameter to keep ID_U secret. U will then derive an encrypted number $N_x = K_{ub} \oplus N_U$ and a validation message $V_1 = h(AID_U||K_{ub}||N_x||S)$. The validation message will be used to verify and continue the phase by checking if V_1 is equal to the hash value with the given parameters. The request message M_1 is created and contains the following $\{AID_U$,N_x, V_1, TS_{ub}, $S\}$, where S is the sensor node U is attempting to communicate with.

Once the request message M_1 is received, B checks TS_{ub} to see if it is valid or invalid by comparing the received TS_{ub} in M_1 with TS_{ub} stored in its database. B then decodes the user's identity and shared key stored in B's database through symmetric cryptography discussed in Section 2.3.

B will verify the AID_U by checking if it is equal to $h(ID_U||K_{ub}||N_U||TS_{ub})$ as well as verifying V_1 is equal to $h(AID_U||K_{ub}||N_x||S)$. Note that N_U was recovered because $N_x = K_{ub} \oplus N_U$ and N_x is sent within the message M_1. Thus only someone who knows K_{ub} can verify N_U thus verifying AID_U. If successful B will know M_1 came from U and will generate a session key SK randomly and a time stamp T. A session key is a type of cryptographic key that is only valid/usable for a limited amount of time [5]. B will encrypt SK such that $SK' = h(K_{bs}) \oplus SK$ where K_{bs} is the shared key between B and the communicating sensor node S; A new validation message will be computed such that $V_2 = h(AID_U||SK'||T||K_{bs})$.

Afterwards B will form and send M_2 containing $\{AID_U$,SK',T,$K_{bs}\}$ to the sensor node S that U wants to interact with. Upon retrieval of M_2, sensor node S will go through a similar verification process checking T, and V_2; thereafter decrypting SK' into SK and forming its own response message M_3:$\{T'$, S, $V_3\}$ to send back to B. M_3 will contain a newly generated timestamp T' and validation message $V_3 = h(SK'||K_{bs}||S||T')$. Lastly B will verify M_3 by checking T' and V_3; once verified B will update its database by changing TS_{ub}, incrementing TS_{ub} by one, and changing its shared keys. B will generate and encrypt a new session key between the user and base station, $SK'' = h(K_{ub}||ID_U||N_U) \oplus TS_{ub}$ and compute the last validation message $V_4 = h(SK''||K_{ub}||N_U||TS)$.

Finally B forms M_4 containing $\{SK''$, TS_{ub}, $V_4\}$ and sends it to U's SC. The SC upon retrieval of M_4 verifies whether the message is valid or not; if valid the SC will store the new values of TS_{ub} and K_{ub}. And finally our user U is authenticated and free to communicate with said node S. In the paper by Gope and Hwang, there is no explicit reasoning behind a user sending a message after being authenticated. Note that if wherever in the process of sending a message M_i to/from B or S, if the process fails to validate the phase will terminate immediately.

3.1.3 Phase 3: Password Renewal Phase

This phase is a password renewal in which, unlike most AA schemes, the user need not communicate with B and is free to change his/her password on the smart card SC. The process is as follows. The user U inserts their identity ID_U, and old password PSW_U and new password PSW_{new} to the smart card SC. SC will retrieve the shared key K_{ub}. SC will create new encoded instances of these parameters using a one way hash function with PSW_{new}. These new instances will replace the old ones.

3.1.4 Phase 4: Dynamic Node Addition Phase

New sensor nodes S^{new} added in the set of sensor nodes need to be taken into account when trying to enhance WSN security. B randomly generates a new unique identity for the the sensor node S_{id}^{new} and new shared key between B and S, K_{bs}^{new}. These keys will be loaded into the sensor node memory by B before being deployed. From now on B encodes K_{bs}^{new} using its unique identity and secret key through a one way hash function. B stores both the new sensor node's unique identity and shared key in its database. All the while B updates the user U with the new information about the node and that it has successfully been deployed, allowing U to access their data. This step helps to solve the security issue of node capture attack mentioned in Section 2.3 and elaborated upon in Section 3.2 by creating distinct shared keys K_{bs} for each individual node in the network.

3.2 Security Analysis

The security analysis will assume that the threat to the WSNs will be an adversary trying to disrupt the network in some manner. In this section we will demonstrate how the proposed 4 Phase AA scheme satisfies multiple security concerns. The proposed scheme accomplishes user anonymity and untraceablity, and resilience against both node capturing and key compromise impersonation attack.

A security issue in WSNs that the proposed 4 phase AA scheme resolves is user anonymity and untraceability. [2] User anonymity and untraceability means that the user, as the source of the data, will remain hidden thus contributing to information protection. This heavily relies on the fact that in Phase 2, during the execution of our authentication protocol, none of the parameters in the request message M_i is allowed to be sent twice. This effectively provides protection against unauthorized listeners.

Security against node capturing is also accomplished in the proposed 4 phase AA scheme. Suppose there is some compromised sensor node S_{comp} and adversary A has obtained information like the private key, and even the session key established between a user and that node. In the proposed scheme, each sensor node shares a unique private key with the base station as well as having a unique session key with the user. This means that an adversary can send false data to a user only from the captured node, not even from B. Other nodes and more entirely the whole network cannot be compromised from the adversary node. This will help ensure the security requirement of data integrity.

One last security issue that is addressed by this protocol is resilience against key compromise impersonation attacks. The impersonation attack was explained in Section 2.2, but we will assume that in this case an adversary A has obtained the server's secret key and the encoded parameters from the server's (B) database. A is trying to impersonate the user in this case. In Phase 1 the user U has identity credentials ID_U such that $ID_U = h(ID_B||\omega||TS_{ub}) \oplus ID_U^*$ where ID_U^* is the encoded identity of stored on B's database. Since one of the parameters to calculate ID_U is the identity of the base station (ID_B) this attack is infeasible because ID_B is information that is baked onto the actual hardware of B. Thus A will have no knowledge of the identity of the base station.

3.3 AA scheme for WBANs

A protocol similar to that of the 4 phase AA protocol is proposed by He et al. [3]; they apply their protocol to a WBAN. The basis of the AA WBAN protocol is Elliptic curve cryptography, a type of public key cryptography [5]. It follows the model described in Figure 2 in Section 2.1.1 where a user or application provider could access a WBAN data through this process. There are three algorithms in this protocol; initialization, registration and authentication. Initialization is used to initialize a connection between the network manager and application provider. Registration is similar to that of Section 3.1.1, where the client will register with the network manager to access data from the application provider. Finally in the authentication step, the client and application provider attempt to authenticate with each other. Once authenticated, the client and application provider will be able to use the WBAN to collect real time data from each other.

A scenario of the applied method is that there's a user with attached sensor nodes measuring his heart rate. The user wants to access this data through mobile, thus needs to go through the process of registering and authenticating with the application provider. This AA WBAN scheme can be applied to this scenario to keep the transit of data secure.

The proposed AA WBAN protocol has provable security, thus there exists a mathematical proof of security discussed in [3]. This proof is based on the fact that the underlying problem is infeasible to solve; this is the Computational Diffie-Hellman (CDH) problem, which cannot be solved in polynomial time. This helps provide resilence against attacks like the replay attack mentioned in Section 2.2 because aspects of the attack would entail solving the CDH problem.

4. SECURITY FRAMEWORK FOR WSNs

A different approach to improve security is to focus on various specific aspects of WSNs to build a solid security framework. This method is different from the other method in this paper, because the other solely focuses on the communication between user, node and base station. The approach of building a framework is taken on by Roy and Nene [7], proposing a framework that will enhance security in military deployed WSNs by focusing on the following elements: Cryptographic algorithm, mode of operation, key management and message authentication code (MAC). In this paper we will focus on the best cryptographic algorithm and MAC in WSNs because of its close relation to the security requirements listed in Section 1.

4.1 Crytpographic Algorithm Used

Confidentiality, a security requirement of WSNs, depends on the cryptographic algorithm chosen to prevent information disclosure to unauthorized users [9]. To choose the "best" cryptographic algorithm we must recall from Section 2.3 that private key cryptography is more appropriate when dealing with WSNs. [7] Private key encryption is divided into two subclasses, block and stream ciphers. In this paper we will primarily concentrate on block ciphers because of numerous problems stream ciphers face in WSNs, as detailed more in [7].

Block ciphers take a block of n bits and convert them to n blocks of encrypted text, where n is the block length. [5, 7] The major difference between block cipher and stream cipher is that block cipher treats data as a collection of fixed size blocks whereas in stream cipher the data is encrypted bit by bit and can be of any length.

To build a strong security framework for WSNs, the block cipher chosen must be lightweight and adhere to the resource restrictions that WSNs face. Lightweight block ciphers in this case mean that they would use 32, 48, or 64 bits as blocks; this will adhere to limited computational capacity and power requirements that WSNs face. *MISTY1*, a 64-bit block cipher with 128-bit keys, was chosen as the block cipher of choice. However, AES-128, a common and popular 128 bit cryptographic algorithm, was heavily considered for the framework. Cryptanalysis, the process of attempting to "break" a cryptosystem, was used to show that both *MISTY1* and AES-128 were secure [5]. *MISTY1* provides the most optimal balance between security level and computational overhead, thus making *MISTY1* the most logical choice when choosing the cryptographic algorithm for this framework [7].

4.2 MAC Specifications

The second specific aspect of this framework that we will concentrate on is the message authentication code (MAC). MAC, also known as a keyed hash function, provides message integrity and message authentication [5]. MAC algorithms are usually used to assure communicating parties that the message hasn't been manipulated in transit.

Choosing MAC deals with two different tasks, message and sender authentication. In the framework proposed by Roy and Nene [7], they focus on picking the "best" algorithm for MAC in WSNs. Best in this case means that the MAC chosen fulfills the security requirements in terms of storage space and computational efficiency. MACs are classified under three different categories, but in this paper we choose to look at a conventional MAC algorithm, the Hash Based Message Authentication Code (HMAC). HMAC is a type of MAC where the key is hashed together with the message [5].

First we will compare HMAC to another MAC also considered for this framework, CBC-MAC (Cipher Block Chaining Message Authentication Code). The way CBC-MAC works increases computational overhead thus deteriorating efficiency, so it is not ideal for WSNs. CBC-MAC works best when the underlying block cipher is 128-bit block cipher, but since the proposed framework uses a 64-bit block cipher, the most compatible choice is HMAC.

HMAC relies heavily upon the underlying hash function used being truly random; in the proposed framework they recommend the MD-5 hash function. MD-5 is a hash function used in internet security for computing checksums of files or for storing of password hashes [5]. MD-5 was chosen because increasing the complexity of the underlying hash function would reduce efficiency in computation [7].

Another reason HMAC was chosen is because it is faster in operations [7]. In the construction of HMAC, the message being sent is only hashed once [5]. So the HMAC construction is thus very low in computation, i.e, faster in operations.

Therefore HMAC provides reliable security in WSNs when paired with the MD-5 hash function and with *MISTY1* as its chosen cryptographic algorithm.

5. CONCLUSION

In this paper we have discussed two different methods of enhancing security in wireless sensor networks (WSNs). Wireless sensor networks face multiple security issues because of resource restrictions. Both the AA protocols and security framework help resolve WSN security issues in their own way. The first method discussed provides a 4 phase (AA) protocol as well as a similar protocol for WBANs. The second is a framework built that focuses on the “best” modules for resource restricted environments like WSNs.

Acknowledgments

Many thanks to Nic McPhee, Elena Machkasova, and Kristin Lamberty for feedback and guidence during the entire process, as well as external reviewer Shawn Seymour for the feedback provided.

6. REFERENCES

[1] What is 2FA? https://www.securenvoy.com/two-factor-authentication/what-is-2fa.shtm. Accessed: 2010-09-30.

[2] P. Gope and T. Hwang. A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 63(11):7124–7132, Nov 2016.

[3] D. He, S. Zeadally, N. Kumar, and J. H. Lee. Anonymous authentication for wireless body area networks with provable security. *IEEE Systems Journal*, 11(4):2590–2601, Dec 2017.

[4] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour. Wireless body area networks: A survey. *IEEE Communications Surveys Tutorials*, 16(3):1658–1686, Third 2014.

[5] C. Paar and J. Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners.* Springer Publishing Company, Incorporated, 1st edition, 2009.

[6] B. Prabhu, M. Pradeep, and E. Gajendran. Military applications of wireless sensor network system. *A Multidisciplinary Journal of Scientific Research Education*, 2(2), 2016.

[7] S. Roy and M. J. Nene. A security framework for military application on infrastructure based wireless sensor network. In *2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 369–376, Nov 2015.

[8] J. H. Saltzer and M. F. Kaashoek. *Principles of Computer System Design: An Introduction.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009.

[9] I. Tomić and J. A. McCann. A survey of potential security issues in existing wireless sensor network protocols. *IEEE Internet of Things Journal*, 4(6):1910–1923, Dec 2017.

[10] T. S. Trana, V. Manh Hoangb, and M. Thang Phamc. A survey for wireless sensor network applications. *The 4th International Conference on Engineering Mechanics and Automation (ICEMA-4)*, 2016.