On the Relevance of Using Multi-layered Security in the Opportunistic Internet-of-Things

Antoine Bagula[1,2(✉)], Lutando Ngaqwazai[2], Claude Lubamba Kakoko[2], and Olasupo Ajayi[1,2]

[1] ISAT Laboratory, University of the Western Cape, Cape Town 7535, South Africa
abagula@uwc.ac.za, 3944991@myuwc.ac.za
[2] Department of Computer Science, University of the Western Cape, Cape Town 7535, South Africa

Abstract. Wireless Sensor Networks (WSNs) have recently gained more importance as key building blocks for the Internet of Things (IoT); a network infrastructure which has greatly increased in number of connected objects with instantaneous communication, data processing and pervasive access to the objects that we manipulate daily. However, WSNs may sometimes need to be deployed in an opportunistic fashion when there is no network with stable power supply available to support the dissemination of the sensor readings from their collection points to a gateway. For such deployments, traditional networking paradigms may fall short to secure the WSNs since most of the well-known security algorithms have been designed for the traditional high quality of service and fully connected networks. Building around some of the security algorithms and protocols which have been developed in the context of Delay Tolerant Networking, this paper presents a multi-layered security model for the opportunistic IoT. The model combines a hash based message authentication code (HMAC) algorithm implemented at the application layer of the IEEE 802.15.4 stack and an Access Control List (ACL) based identity based encryption algorithm used by the IEEE 802.15.4 MAC layer as a new and novel method of signing and authenticating data which is stored and forwarded on an opportunistic Internet of Things (IoT) infrastructure.

Keywords: Multi-layered security · Internet-of-Things · Wireless sensor networks · Opportunistic networking

1 Introduction

Recently, Wireless Sensor Networks (WSNs) have gained more significance as key building blocks for the Internet of Things (IoT). As a network infrastructure, it has seen a great increase in the number of connected objects with instantaneous

R. Zitouni et al. (Eds.): AFRICOMM 2019, LNICST 311, pp. 15–29, 2020.
https://doi.org/10.1007/978-3-030-41593-8_2

communication, data processing and pervasive access to objects that we manipulate daily at "anytime", from "anywhere" and using "anything". Next generation wireless sensor networks are predicted to deployed on the *Internet-of-the-Things (IoT)* paradigm, connecting islands of ubiquitous sensor networks (USNs). Some of these USNs would follow an opportunistic communication model where the sensor motes are provided the capability of communicating with each other even in the absence of established route(s) between them. This is achieved by building communication pathways "on-the-fly" by having each message finding its next-hop to a gateway opportunistically, with the expectation of bringing the message closer to its USN gateway. When deployed in the harsh environments of the developing world where grid power supply is intermittent or energy is harvested through solar, wind or other energy scavenging methods, these islands of USNs provide a great opportunity to positively impact different daily life activities and bridge the digital divide as they are built around low cost, cheap-to-maintain devices; making it easy to construct networks that can be deployed unattended in thousands of nodes. A typical opportunistic IoT networking scenario is depicted in Fig. 1, where pollution sensing devices are deployed in a city with the need to measure different types of pollution. There is the need to have different settings for the sensing node in order to accurately measure the pollution levels at various areas of the city. In essence thee is the need to form a pollution monitoring network with settings tuned for the particular application area viz.: industrial pollution network in industrial settings, education pollution network in educational settings, residential pollution network in residential settings and traffic pollution network on roads. In such networks, buses, humans and bikes can be used as mobile gateways collecting data opportunistically from the pollution sensors when they are within communication range of these sensors. A similar scenario can be considered in rural deployment where a rural sensor mesh network can benefit from a mobile network of humans, buses and bikes playing the role of data collectors for the sensor network. These scenarios illustrate typical opportunistic IoT deployments in both developed and developing regions when reliable end-to-end network solutions are either financially not feasible or unavailable, and would possible not be in the foreseeable future, due to problems not limited to unsustainable power supply and/or poor local infrastructure. As suggested by the authors in [1], opportunistic networking is more general than delay tolerance, although both terms are often used interchangeably. While opportunistic networking systems are a viable low entrance solution to connecting devices over vast distances, they are still a green field in terms of field readiness and for research in the security field. Figure 2 reveals the interactions between the data mules and the IoT network infrastructure.

1.1 Related Work

Research works on opportunistic networks' security have mostly been conducted in the context of delay tolerant networks. Kate *et al.* [12] showed that security and anonymity are critical in many delay tolerant network implementations and

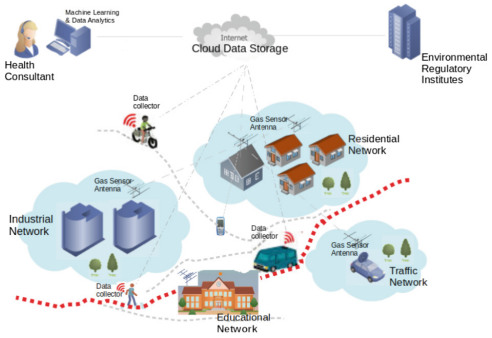


Fig. 1. The opportunistic IoT network infrastructure

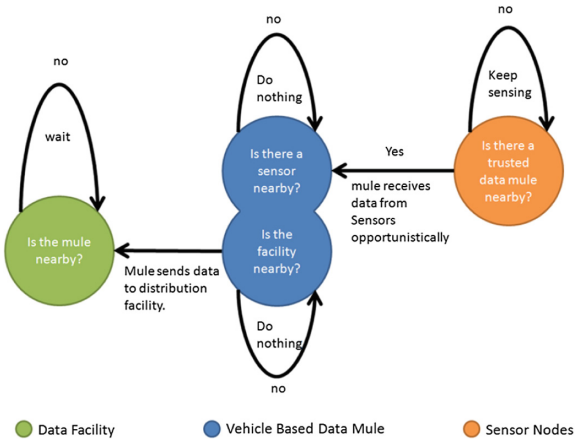


Fig. 2. The mule-infrastructure interaction model

revealed that due to the very low quality of service nature of delay tolerant networks, traditional security implementations based on public key cryptography and infrastructure are not suitable for these intermittent networks. They also showed how maintaining anonymity in delay tolerant networks is not applicable either and demonstrated how identity-based security (IBS) can be used as a security architecture suitable for such network implementations. While their work made a deep analysis of key revocation and generation, our work falls beyond the scope of such security mechanisms. In [13], Seth *et al.* addressed the challenges for securing data communications in delay tolerant networks by making various valid points about how traditional public key infrastructure (PKI) based approaches are not suitable for such networks. In a PKI, a user authenticates another user's public key by using a certificate signed by a certificate authority. In highly intermittent network infrastructures like a delay tolerant network, gaining instant access to a trusted third party certificate authority can prove to be infeasible due to the disconnected nature of the network. PKIs also do

not work because key revocation cannot be enforced in an instance. This results in a situation where nodes with compromised keys or outdated keys cannot be forcefully removed from self authenticating them. In a book on delay tolerant networking written by Farrell and Cahill [14], the use of IBS in a delay tolerant network is heavily criticized for not having a valid key management solution as it falls short of providing proper authenticity. This is exemplified by the fact that in DTN environments, it would be very hard to discern between two "Bobs" as the second bob may be an attacker or intruder. This problem can be overcome by using IP or MAC addresses in conjunction with a unique device identifier as the device identity. This however, does not stop an attacker from assuming a name which is the same as another node on the network by using IP/MAC spoofing. Farrell and Cahill also believe that IBC-based security is not scalable as the user must know the public parameters of all the private key generators (PKGs).

1.2 Contributions and Outline

This paper's main contribution is to propose and assess the relevance of using a multi-layered security model that harnesses some of the features of the existing 802.15.4 mac layer security and uses a novel application layer security algorithm to secure communication over a wireless sensor network. The security approach adopted in this paper combines a Hash-based Message Authentication Code (HMAC) algorithm with an Identity Based Encryption (IBE) algorithm to secure an opportunistic network of wireless sensor nodes. Using prototyping, the model was implemented on an arduino-based wireless sensor device as a new and novel method of signing and authenticating data which is stored and forwarded on the opportunistic network. This method involves using the MAC layer security features in the IEEE 802.15.4 specification and an application layer security implementation. The method proposed in this paper has been designed to secure healthcare networks in the context of a CyberHealthcare project described in [5–8] and [9,10]. The project aims at collecting pollution data in areas of interest and correlating the data to health issues in these areas.

The remainder of this paper is organized as follows. Section 2 presents the proposed multi-layered security model. The results of the cyberhealthcare field readiness experiments in the city of Cape Town are presented in Sect. 3 while Sect. 4 contains our conclusions.

2 The Multi-layered Security Model

Some of the main goals of a security model are to provide data integrity and authenticity and data confidentiality. These goals have been used to guide the design and implementation of the multi-layer security model. It involves utilizing the existing technologies contained in 802.15.4 and other features resulting from security requirements that cannot be met by using the security features of the standard only. These include application layer cryptography and message

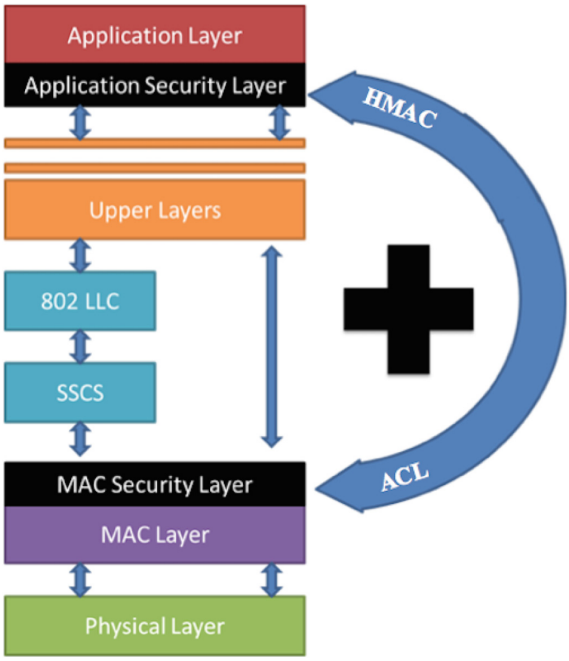


Fig. 3. The multi-layered security model

authentication codes which need to be implemented to add additional security safeguards against spoofing, forgery and service attacks on the delay tolerant network. As depicted by Fig. 3, the multi-layered architecture reveals different layers of a typical 802.15.4/ZigBee protocol but also highlights the two layers where the proposed multi-layer model is implemented, that is: (a) at the MAC security layer where access control lists are used to enable only authorized data mules to interact with the system and (b) at the application security layer where encryption/decryption mechanisms are implemented to secure data.

2.1 Data Integrity and Authenticity

Cryptographic hash functions are usually used for data integrity and authenticity. A cryptographic hash function is an algorithm which takes an arbitrarily sized block of data (usually called the message), and returns a fixed size fingerprint or signature of the data (usually called the digest) [2,3]. Cryptographic hash functions are characterized by the following key features:

- Modifying the message without changing the hash is not feasible.
- The hashing of a message is easy.
- Reversing the hash process is not feasible.
- Two different messages must not hash to the same digest.

The hash function itself is publicly known, thus allows for any mote to implement and use it.

Integrity check: When a message is sent within the network, it will usually arrives in two parts, the first n bits would be the hash (or digest) of the original message [2], while the last m bits (appended to n) would be the actual message. To check that the integrity of the data is intact, a mote will hash the original message, and compare the digest of the message which the mote calculated, to the digest which was sent over the network. This ensures that no tampering of the data was done while in transmission. Such an integrity check can safeguard against man-in-the-middle attacks, and can also be used as a rudimentary error-detection method. Observe in Fig. 4, if two motes are communicating and mote A sends mote B the communicated data (which contains the digest and message appended to it), Mote B can calculate the digest of the message bit string and do a bitwise comparison between its own calculated digest and the digest which was sent by mote A.

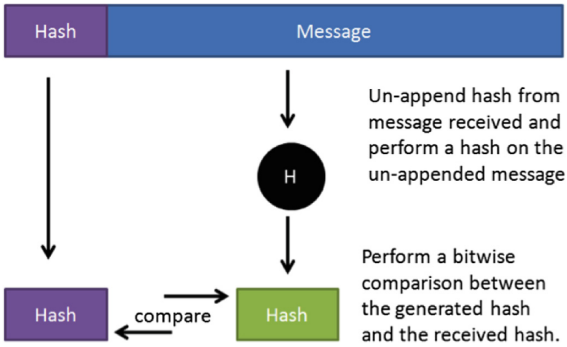


Fig. 4. Integrity check illustration

Hash-Based Message Authentication Code (HMAC): A hash-based message authentication code is a type of cryptographic construct which allows the calculation of a message authentication code using a cryptographic hash function [2]. To generate an HMAC, two pieces of data are requires - a key and the message itself. HMAC is commonly used to verify the authenticity of data and at the same time serves as a data integrity check mechanism (message error check). HMAC generation is defined by:

$$HMAC(K, m) = H((K \oplus o_pad) \mid\mid H((K \oplus i_pad) \mid\mid m)) \quad (1)$$

where o_pad and i_pad are the outer padding and inner padding respectively.

2.2 Data Confidentiality

802.15.4 is an IEEE standard which offers functionality of the lower layers of the OSI network model. It is a very lightweight Wireless Personal Area Network

(WPAN) specification which ensures that the cost of producing an 802.15.4 radio is cheap. 802.15.4 is slow (250 kb/s) and covers low range (10 m), though the range can be increased by tweaking various power output values of the radio transceiver. Collision avoidance follows the CSMA/CA scheme. The 802.15.4 secure communication works mainly on the media access control (MAC) sub layer which offers protection of the payload using AES 128 encryption. Rudimentary integrity and code error checking is also performed when communication is being done.

Security Modes. The 802.15.4 allows for various modes of communication. These modes of communication are typically set by the application utilizing the medium access layer protocol. There are a multitude of modes and they will be discussed. The media access control layer consults or sets the flag field in the packet to determine or set the security mode which must be applied to the communication transaction. If no security flag is set, then the packet is passed onto the application layer as is. If there is a security flag checked, then the media access control layer will conform to a protocol which discerns what type of security methods should be employed for that particular communications security flag [18].

No Flag Set: This is the simplest setting that you can have on 802.15.4. It does not perform any additional security functionality and provides zero security guarantees [11].

AES-CTR: This security mode provides confidentiality using AES with counter mode. Basically the sender partitions the plaintext packet into 16-byte blocks. For every plaintext packet, there is an equivalent cipher text packet.

AES-CBC-MAC: This security mode provides integrity assurances using CBC-MAC. The sender can compute a MAC using a CBC-MAC algorithm. The MAC can only be computed by parties with the same symmetric key. The MAC basically encapsulates packet headers and the data payload (which comes from the application). The sender appends the plaintext data with the MAC, and the recipient verifies the MAC by re-computing the MAC by reconstructing the packet and computing the MAC again against the sent MAC [18].

AES-CCM: This security mode uses CCM mode for encryption and authentication. It initially applies integrity protection over the header and data payload using the CBC-MAC algorithm, it then encrypts the data payload and MAC using the AES-CTR mode. One can think of AES-CCM as the combination of CBC-MAC and AES-CTR modes [18].

Security in IEEE 802.15.4: Payload encryption and access control lists are discussed in this section as 802.15.4 security mechanisms.

Payload Encryption: The payload (or message) can be encrypted in three different ways, depending on your network configuration and network requirements. The payload can be encrypted using AES-CTR scheme, which is just

an encryption on the payload but the frame counter sets a unique message ID. The AES-CBC-MAC scheme is when the message authenticity code (MAC) is appended to the data payload. This adds authentication and integrity checks at the medium access control layer. And then there is AES- CCM, which is a combination of the two previously stated schemes.

Access Control Lists (ACLs): The Access Control List (ACL) is a list of "trusted devices" along with the security policy that each of the devices have. Each node must have an ACL. An ACL is a table containing the nodes address (IP and MAC address pair), the encryption algorithm allowed by that node (AES-CTR, AES-CCM etc.) and the Replay Counter field, which is used to avoid replay attacks. When a node wants to send or receive a packet to or from a specific device, it checks the packet, to see if it from/to a node in the ACL of the source/destination. If the device is a trusted unit, an authentication process will take place.

2.3 The Proposed Security Solution

With only 8 KB of usable primary memory, the Wasp Mote is a low power sensor device running an 8-bit CPU and a 16-bit ALU clocked at 8 MHz. It uses a radio transceiver which conforms to the IEE 802.15.4 specification. This specification mandates that 802.15.4 radios have AES-128b and error checking capabilities. It also requires hardware managed Symmetric-key key cryptography, which can be set to either pair- or group-wise. The radios also have a feature called access control lists allowing devices with only certain MAC addresses to be permitted on the network.

2.4 The Key Exchange Algorithm

As presented in Fig. 4 and discussed above, the key exchange algorithm works similarly to that which is present in today's world of digital signatures. Basically if two entities on the network wish to communicate, then both of the entities need to assure themselves that both of the nodes are infact authorized nodes within the network. Public key cryptography presents the best option for entities on the network to mutually authenticate each other. However, since public key cryptography could not be implemented, a different approach using Access Control List (ACL) model had to be taken. This way, each node has to make sure that they are on each other's ACL. The ACL is defined before network deployment, and can be updated over the air. If the nodes are mutually trusted devices then shared key exchange can take place using a key agreement scheme called the symmetric-key key exchange. This algorithm is depicted in Fig. 5.

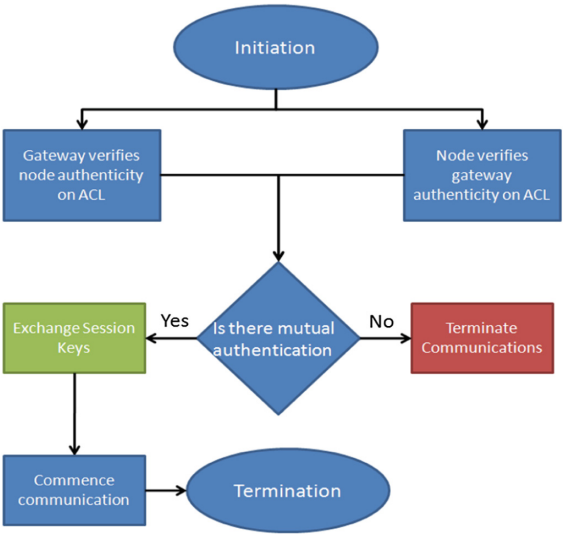


Fig. 5. The key exchange algorithm

3 Performance Evaluation

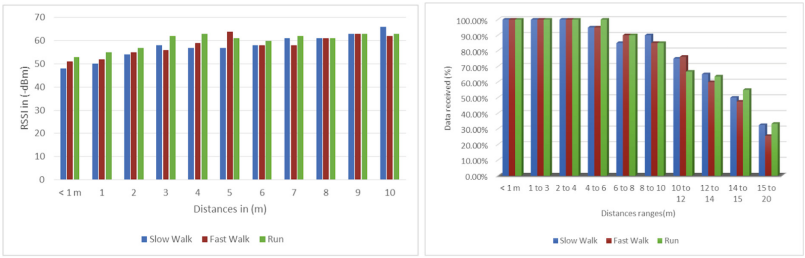
Different experiments were conducted to assess the field readiness of opportunistic networks when considering the distance between communicating nodes and the security features implemented using the multi-layer security model.

3.1 Opportunistic Contacts

Opportunistic contacts define moments when two devices (mobile nodes) of an IoT network can exchange messages with mesh router at acceptable signal strength and throughput. To assess the field-readiness of the oppotunistic IoT scenario described in this paper, we conducted a number of experiments to determine how the communication range and the speed of the mobile node can impact the opportunistic contacts. The results presented in Fig. 6 reveal that:

- both the RSSI and throughput decrease as distance increases.
- while the throughput for different modes seem not significant, the RSSI may have a different impact on the quality of signal received as these values are expressed in dBm.

Further investigations are being conducted to assess the quality of the opportunistic contacts for cars and UAV drones, and to comparing obtained results from these, with those presented in this paper.



(a) RSSI at different ranges. (b) Throughput at different ranges.

Fig. 6. Opportunistic contacts.

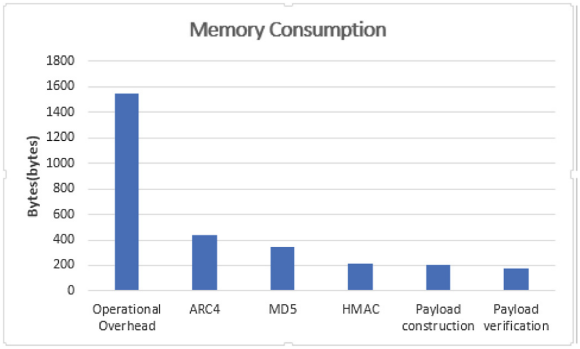


Fig. 7. Memory usage

3.2 Impact of Security on Memory Usage

Figure 7 reveals that the secondary memory footprint of the application security layer did not exceed 2 KB. Typically, devices using 802.15.4 as a means of communication hardly have over 24 KB of capacity on the programmable EEPROM. In terms of primary memory, the application security layer did not exceed 2 KB in usage although it occupied 1.5 KB of the device's RAM. The operational overhead was lower because not all items in memory were explicitly implemented and it was measured at the beginning of each loop. These operational overheads usually include processes which handle the API subroutines, interrupt management or allocated hardware memory allocation. Arduino uses both memory mapped I/O and bus/hardware mapped I/O. The ARC4 algorithm used the most amounts of bytes in memory, consuming about 446 bytes. A large chunk of the bytes can be attributed to the permutation array "S" which is of the size 256 bytes. The rest can be attributed to the cipher text array (The cipher text array is the exact same size as the plaintext array) The MD5 algorithm consumed about 326 bytes of memory. MD5 characteristically chunks a message or plaintext into chunks of 64 bytes (pads the message with zeros to make it naturally "chunk-able" by 64 bytes). The other byte consumed can be

attributed to other arrays like the message digest context. The HMAC algorithm uses 224 bytes. This measurement was obtained by subtracting the MD5 byte usage from the total HMAC byte usage. The HMAC uses a few memcopy calls to store the key outer and inner padding arrays. Payload Construction is the routine which is responsible for allocating memory space enough to append a 32 byte hash to a payload of 60 bytes (give or take) and then encrypt it using the ARC4 stream cipher. The ARC4 stream cipher text is double the size of the plaintext since hexadecimal encoding is used to encode bytes and two hexadecimal values are needed to encode a single byte.

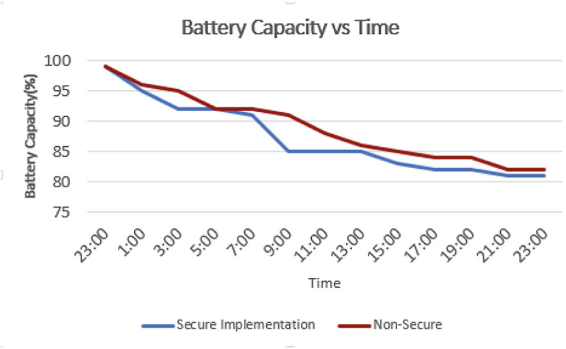


Fig. 8. Battery usage

3.3 Impact of Security on Battery Usage

The power overhead for securing communications can be seen in graph depicted in Fig. 8. This data shows that the power drain resulting from our multi-layered model is slightly higher compared to a non-secure implementation of 802.15.4 communications. The battery usage test was conducted over a 24 h period. Libelium have lauded their devices' minimal power requirements and have openly marketed that their product can be battery powered for up to 1 year with the correct hibernation and sleep programming practices. This experiment was conducted by having a Wasp Mote using 802.15.4 send arbitrary packets to a gateway node approximately 3 m away from it. The experiment was conducted by using the deep sleep method. Each time the wasp mote woke up, it would send a Comma Separated Value (CSV) of its current battery to the gateway. CSV was used in order to make plotting the data easy on tools such as MS Excel. This battery test was done for two cases, the first involved sending data to the gateway using the security methods; while in the second case, data was sent in plaintext.

3.4 Impact of Security on Processing Time

As shown by Fig. 9, the processing time is lower for the MD5 security encryption while payload construction consumes more processing time followed by payload verification. It can be observed that the processor is always at 100% usage. This is in agreement with most Arduino boards because they are typically 95%–100% in use at all times as a result of the processor itself having a very low clock speed. Even if the device had minimal code flashed onto it (The Hello World example code) the processor usage rarely drops below 95%. The Arduino community at large has unanimously experienced this. The graph in Fig. 9 details the processing times of the various routines which make up the bulk of the security methods. The data input size was the maximum allowed data payload allowable by the 802.15.4 specification (96 bytes). The payload construction and payload verification routines often make use of the HMAC-MD5 and ARC4 algorithms, this explains their relatively higher execution times. However, it should be noted that ARC4 only takes about 20 ms to encrypt a payload of size 96 bytes. We noticed that the payload construction is high because it requires the device to do dynamic memory allocation many more times than the Payload verification algorithm would need to do.

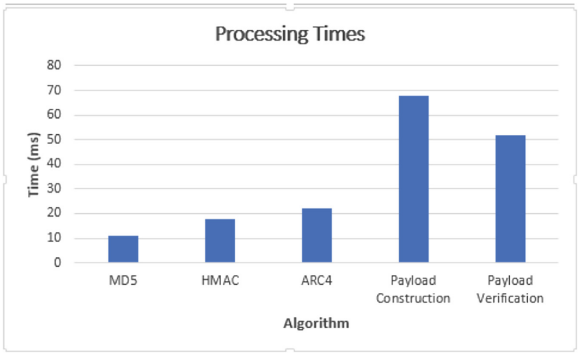


Fig. 9. Processing time

3.5 Impact of Security on Hash Function Overheads

Figure 10 depicts the hash function overheads in terms of execution time of the hash function and the HMAC routine. The results reveal that the HMAC-MD5 model outperforms the others in terms of processing time. The stand alone MD5 function is known to be a cryptographically insecure algorithm, and has thus lost its appeal as a hash function. SHA-1 is generally regarded as being a safer solution. However, in this work it was combined with the HMAC therefore making it safer and less susceptible to attacks. MD5 was chosen as the underlying HMAC function because it is the lightest hash function known.

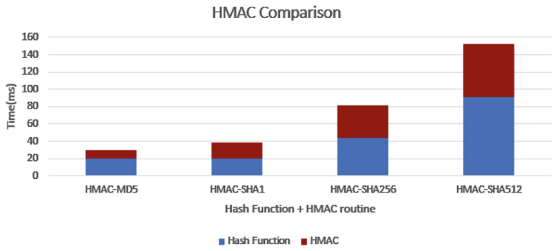


Fig. 10. Hash function overheads

Figure 10 shows the relative difference in execution time between the HMAC portion and the the underlying hash function. It reveals that HMAC-MD5 is indeed very light weight. When the other hash functions were run on the wasp motes, they all required more execution time. This is due to the fact that MD5 (used in HMAC-MD5) produces the smallest hash and works on the smallest block size. Of significant note, is that MD5 is not necessarily a compromise as far as hash functions in HMACs are concerned, as it is secure enough as a means of achieving message authentication and integrity checks.

4 Conclusion and Future Work

This paper has shown that a solution which satisfies security requirements for delay tolerant network of light-weight devices is an achievable goal. This paper also shows how lightweight cryptographic methods can be used to secure a delay tolerant network and revealed how two different protocol layers can be used to achieve network security. The multi-layered approach shows that coupling the security features in 802.15.4 and current lightweight cryptography is a great way to secure a network. The results extracted from a number of experiments conducted around the city of Cape Town to test the field readiness of a cyberhealthcare network infrastructure reveal that our multi-layered security algorithm is a viable security solution that ensures (1) data confidentiality, authenticity and integrity and (2) a lightweight implementation solution which is suitable for the low processing and memory footprint of the low powered sensor devices. The proposed solution had minimal impact on the device's primary memory as it only used a total of just below 1 KB of memory (out of 8 KB) and used a minimal amount of secondary memory (12 KB out of 128 KB).

The results presented in this paper also show that implementing this solution on lightweight devices imposes minimal memory, low processing footprint but requires marginally more battery power.

The management of the opportunistic IoT infrastructure is a key parameter that may require redesigning existent network management techniques to efficiently engineer the cyber-healthcare system. When redesigned in the context of opportunistic networking, the multipath routing techniques such as presented

in [15,16] may be used to support QoS by having different forms of healthcare data propagated over different paths form a source to a destination. These methods can be combined with the redesign of the cost-based traffic engineering techniques proposed in [17,18] to balance traffic over the cyber-healthcare communication platform and thus increase throughput and reduce communication delays. Extending the reach of an opportunistic IoT infrastructure over a long distance wireless mesh sensor network is another key issue that needs to be addressed that needs to be addressed to support cyber-healthcare network deployment in the rural settings of the developing world. These management techniques are an avenue for future research work.

References

1. Pelusi, L., Passarella, A., Conti, M.: Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. IEEE Commun. Mag. **44**(11), 134–141 (2006)
2. Kaps, J.-P.: Cryptography for ultra-low power devices, Ph.D. thesis, Worcester Polytechnic Institute (2006)
3. Aumasson, J.P., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: a lightweight hash. J. Cryptol. **26**(2), 313–339 (2013)
4. Padmavathi, G.: A survey of attacks, security mechanisms and challenges in wireless sensor. Networks **4**(1), 1–9 (2009)
5. Mandava, M., Lubamba, C., Ismail, A., Bagula, H., Bagula, A.: Cyber-healthcare for public healthcare in the developing world. In: Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina-Italy, 27–30 June 2016, pp. 14–19 (2016)
6. Bagula, M., Bagula, H., Mandava, M., Kakoko, C., Bagula, A.: Cyber-healthcare kiosks for healthcare support in developing countries. In: Proceedings of the AFRICOMM 2018, Dakar-Senegal, 29–30 November 2018
7. Celesti, A., et al.: How to develop IoT cloud e-health systems based on FIWARE: a lesson learnt. J. Sens. Actuator Netw. **8**(1), 7 (2019)
8. Bagula, A., Mandava, M., Bagula, H.: A framework for healthcare support in the rural and low income areas of the developing world. J. Netw. Comput. Appl. **120**, 17–29 (2018). https://doi.org/10.1016/j.jnca.2018.06.010
9. Bagula, A., Lubamba, C., Mandava, M., Bagula, H., Zennaro, M., Pietrosemoli, E.: Cloud based patient prioritization as service in public health care. In: 2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT), pp. 1–8. IEEE (2016)
10. Lubamba, C., Bagula, A.: Cyber-healthcare cloud computing interoperability using the HL7-CDA standard. In: 2017 IEEE Symposium on Computers and Communications (ISCC), pp. 105–110. IEEE, July 2017
11. Murillo, M.J., Aukin, M.: Application of wireless sensor nodes to a delay-tolerant health and environmental data communication system in remote communities. In: Proceedings of the 2011 IEEE Global Humanitarian Technology Conference, pp. 383–392, October 2011
12. Kate, A., Zaverucha, G.M., Hengartner, U.: Anonymity and security in delay tolerant networks. In: Proceedings of the 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007, pp. 504–513 (2007)

13. Seth, A., Keshav, S.: Practical security for disconnected nodes. In: Proceedings of the 1st IEEE ICNP Workshop on Secure Network Protocols, pp. 31–36 (2005)
14. Sastry, N., Wagner, D.: Security considerations for IEEE 802.15.4 networks. In: Proceedings of the 2004 ACM Workshop on Wireless security - WiSe 2004 (2004)
15. Bagula, A.B.: Modelling and implementation of QoS in wireless sensor networks: a multi-constrained traffic engineering model. EURASIP J. Wirel. Commun. Netw. **1**, 1–14 (2010)
16. Bagula, A.B.: Hybrid traffic engineering: the least path interference algorithm. In: Proceedings of the 2004 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, pp. 89–96, South African Institute of Computer Scientists and Information Technologists (2004)
17. Bagula, A.B.: Hybrid routing in next generation IP networks. Comput. Commun. **29**(7), 879–892 (2006)
18. Bagula, A.B.: On achieveing bandwidth-aware LSP/LambdaSP multiplexing/separation in multi-layer networks. IEEE J. Sel. Areas Commun. **25**, 987–1000 (2007)