CSCM88: Network and Wireless Security Mini Project

MSc Advanced Computer Science

Multicast Authentication in Internet of Things

Luke Hengstenberg

878876

Nathan Grimble

876039

Supervised by

Dr. Pardeep Kumar
Department of Computer Science

30 April 2020

Contents

Abstract

Multicast communication is essential for improving performance in the constrained environments used by IoT devices. Without a cryptographically secured authentication protocol multicast communication between IoT devices is vulnerable and susceptible to attack. A wide range of security mechanisms have been proposed by the scientific community but choosing between them can be challenging due to differences in performance, application and security. This report selects four multicast authentication protocols and conducts an in-depth analysis of design, performance, achievements, and limitations. This report aims to discover the protocol that best achieves continuous multicast authentication for IoT applications using a secret sharing scheme. Through the paper's own analysis and the reporter's comparative analysis a conclusion is reached that the paper "Symmetric-Key-Based Security for Multicast Communication in Wireless Sensor Networks" [4] best satisfies this motivation. Furthermore, this report suggests enhancements to the research based on discovered weaknesses.

1 Introduction

The Internet of things (IoTs) is the collective concept that devices connected to the internet have the ability to communicate with each other without the need for human interaction. Multicast communication greatly improves performance in the resource constrained environments used by IoT devices as a single message can reach a group of nodes simultaneously. Over the last decade, the number of IoT devices has significantly increased. As of 2018 the amount of IoT devices is estimated to be 22 billion [1]. The number of IoT applications and devices are constantly increasing and have been implemented in almost every location from smart homes to healthcare to agriculture [2]. Communication between IoT devices often contains sensitive information meaning privacy and security is critical. Without adequate protection multicast communication can be extremely vulnerable, enabling theft of data and other attacks to be mounted against IoT devices. As a result of this a sophisticated multicast security mechanism is essential to protect the confidentiality of communication, integrity of sent/received data, and authenticity of the communicating nodes. Due to the commonly restrictive nature of IoT devices (Different environments / resource constraints) the mechanism needs to be lightweight and independent of the environment in which the devices are in. Research into multicast authentication for IoT is well documented and a wide array of approaches have been devised by the scientific community. However, selecting the best multicast authentication mechanism can prove challenging as certain approaches contain unconsidered performance and security issues.

This report aims to find a continuous multicast authentication protocol for IoT applications implementing a secret sharing scheme. This report reviews several multicast authentication protocols, analysing the differing approaches to secure key management and applicability for low bandwidth multicast communication in IoT environments. Section 2.1 contrasts the problem definitions and motivations fuelling each research piece, identifying those in fitting with this project's motivations. Section 2.2 delves into the structure of each protocol and chosen methodology, summarising the key components, and drawing comparisons to discover strengths and weaknesses in design. Section 2.3 breaks down the key findings and experimental results obtained from simulation and performance analysis of the protocols, highlighting those with superior efficiency and security, as well as finding aspects that could be improved. Section 3 discusses the broader significance and real-world application of the contributions, conducting an in-depth analysis of what was achieved and using weaknesses to propose enhancements to the research.

2 Multicast Authentication

2.1 Research Motivation

When investigating academic publications proposing multicast authentication protocols a common problem definition can be deduced. All researchers feel the resource constrained and lightweight nature of the IoT environment poses the biggest challenge due to the limited capacity for implementing the necessary security mechanisms. Yao, et al. [3] stress the vulnerability of multicast communication in IoT and the need for a lightweight authentication mechanism protecting multicast messages. Carlier, et al. [4] feel multicast communication is a necessity for improving performance in constrained environments and agree that a security mechanism must provide at

least the features of confidentiality, integrity, and authenticity. Both Park [5] and Bamasag and Toumi [6] reinforce the benefits of multicast communication for reducing bandwidth requirements and improving efficiency of communication, as well as aiming to address the challenges of maintaining reliability and security. Each publication mirrors this project's motivation to some degree and shares the broad objective of securing multicast authentication in constrained environments. However, clear differences can be seen based on novelties in approach and individual motivation to extend existing research. Personal motivations include extending Nyberg's fast one-way accumulator [3], proposing a new multicast key management protocol for wireless sensor networks (WSNs) [4], improving the security and performance of multicast communication for constrained application protocol (CoAP) applications [5], and extending an existing continuous authentication protocol to support multicast [6]. The prior research forming the basis of each approach creates clear differences in design which will become apparent in the succeeding sections. Moving forward, this paper will continuously question whether the primary motivation of implementing a secure multicast authentication protocol for IoT applications based on a secret sharing has been achieved.

2.2 Overview of Protocols

One approach to creating the multicast authentication mechanism was developed within the Institute of Electrical and Electronics Engineers (IEEE) [3]. Yao, et al. [3] analysed and revised Kaisa Nyberg's "fast one-way accumulator" [7], an accumulator for cumulative hashing that utilises existing hash algorithms and pseudorandom sequence generators. It was noted that the accumulator's properties of "absorbency", "one-way hashing" and "quasi-commutative" were ideal for applications in which gathered items are dynamic, therefore modifications were made to support multicast authentication, simplified ease of processing and lightweight functionality for small scale resource-constrained IoT devices. Nyberg's fast one-way accumulator utilises no trap door itself, but rather the abstract general hash function and simple computational operations to achieve fast accumulative hashing. The security protection from Nyberg's fast one-way accumulator is dependent on the complexity of creating an accumulated hashed item successfully and the randomness of the produced hash from the hash function. Yao, et al. [3] revised Nyberg's accumulator by embedding MACs (Message Authentication Code) within the accumulator to perform multicast authentication directly on the multicast data. The multicast data consists of two parts, one being a shared key between the source and recipient, the other being the multicast data to be communicated. The actual hashing function makes use of Hashed-Based Message Authentication Code (HMAC) substituting the hashing function h in the original algorithm. The strengths of this scheme are its sophisticated verification mechanisms ensure signatures always correspond to those of the sender and receiver, the hashing process is cryptographically secure and tamper resistant, and extensive five-step authentication must always be passed.

Carlier, et al. [4] proposed a second approach to multicast authentication with the design of a security scheme including a novel key management protocol facilitating secure multicast communication for wireless sensor networks (WSNs). Unlike [3], [4] details an entire multicast authentication protocol rather than an authentication mechanism based on assumptions of incorporation within a working system. [4] organises and manages multicast communication and groups through the incorporation of a cloud-based network multicast manager (NMM) based on the previous

work of one of the researchers, Akkermans, et al. [8]. For the multicast engine the researchers use the multicast protocol "Bidirectional Multicast RPL Forwarding (BMRF)" also taken from their existing research [9]. Use of the NMM architecture allows multicast groups to be integrated with a publish/ subscribe pattern, utilising known advantages backed by existing research such as Traintafillou and Economides [10], where improved network performance is reported due to the pattern's reduced bandwidth usage and minimal storage and computational requirements. Moreover, use of the BMRF protocol integrates the features and respective advantages detailed in [9], including added dynamicity to group registration, multipoint-to-multipoint communication (multiple senders), bidirectionality, and options for both multicast and unicast. Carlier, et al. [4] establish five phases in the security scheme which are: 1) Key distribution, 2) Registration, 3) Group Key Construction, 4) Multicast Communication, and 5) Group Key Update.

In phase 1), the Trusted Third Party (TTP) generates security information for the nodes and the NMM, sending over a secured channel. All parties are given unique identities and cryptographic hash functions for authentication therefore preventing imposters in a similar method to [3]. In phase 2), the security information from phase 1 is used so the authenticated nodes can communicate directly with the NMM to share interests and establish a common shared secret key. Integrity checks are used upon the arrival of each message, comparing sending/receiving timestamps to end communication if too much time has passed, or rejecting messages if a value has changed. In phase 3), multicast groups are created based on phase 2 with the NMM sending subscriber key information to the publisher (P) to create a common shared key between P and each subscriber, which is used by P to create and distribute a group key using a combination of Lagrange interpolation (LI) and an IP multicast packet. A one-way key chain is constructed to authenticate all communication and a secret sharing is employed. The authentication mechanism in this phase shows compatibility between [4] and [3] through incorporation of the revised one-way accumulator, presenting an interesting avenue for future research. In phase 4), the shared group key from phase 3 is used for the publisher to securely communicate with subscribed nodes, achieving the desired multicast communication. In phase 5), The group key is updated in circumstances where a node joins or leaves a multicast group.

Park [5] developed an extension to the security and performance of multicast communication with the constrained application protocol (CoAP). Much like [4], a key management framework was developed, however instead of building on BMRF, [5] aimed to secure DTLS-based multicast CoAP and improve its performance by replacing the DTLS Handshake. Furthermore, instead of a publish/subscribe model [5] used the request/response client/server architecture. Park [5] makes use of a resource directory (RD) server for the storage of CoAP server information and the lookup functionality for CoAP clients, and a group controller (GC) to provide security credentials and control group membership. Similarities can be seen between the broad role of the GC and the network multicast manager (NMM) used in [4], however these are trivial. In comparison to [4] the key management phases are not clearly defined; nevertheless, the scheme's design can be split as follows:

1) RD server holds copies of resources on CoAP servers and facilitates GC functionality. 2) GC allocates an IPv6 address and maintains a group of CoAP servers. Performs registration, distribution and rekeying phases with each server. Enforces access control for each CoAP client.

3) A registration protocol is designed based on Brown, et al. [11] ECQV certificate issuing. GC is certificate authority issuing ECQV certificates to CoAP clients and servers using public-key cryptography. In this protocol ECDH keys are established between clients and GC (unicast key), and servers and GC (multicast key). The strengths of this protocol are that the keys are cryptographically strong due to them being generated by the elliptic curve of large prime divisors, communication is secured by TLS/DTLS, and signatures are generated to maintain integrity. 4) After registration is successful CoAP clients exchange Request/Response messages with servers to initiate CoAP applications. In the first message exchange a key is established between clients and servers. Clients use the multicast server group public key and the clients private unicast key to compute a new group key, enabling CoAP request messages to be sent from a single client to a group of CoAP servers. Following this, each server computes the private unicast key and sends a CoAP response message to the client. The strengths of this protocol are CoAP messages can be sent securely because unicast and multicast keys have been established between clients and servers. Furthermore, information can be encrypted whereas previously this CoAP communication would be insecure.

The final approach was developed within the Department of Mechanical Engineering of the Massachusetts Institute of Technology [6]. Much like [3], Bamasag and Toumi [6] detail a multicast authentication mechanism rather than a complete functioning multicast system as in [4] and [5]. The mechanism aims to provide secure and efficient multicast authentication for environments where message communication must be frequent with minimal overhead. The mechanism is built on a symmetric central secret sharing scheme called "Shamir's secret sharing scheme", which like [4] takes advantage of symmetric cryptographic efficiency and minimal communication requirements in comparison to asymmetric solutions. [6] establishes a secure communication channel between the source and recipient end-nodes for a specified period, labelled "continuous authentication" [12] a previous project implemented within the proposed protocol. Utilising "Shamir's secret sharing scheme" originally proposed by Shamir in 1979 [13], the main purpose was to create a specific number of secret shares and distribute them to an equal number of shareholders. When these authorized shareholders communicate with each other, they can retrieve the origin secret and authenticate the sender and the message's integrity. Bamasag and Toumi [6] utilised the secret shares for a different purpose as authenticating tokens instead that enable the authentication of the sender node to the recipient end-node within a specific timeframe. This permits the recipient end-node to trace the obtained secret share back to its origin secret and therefore the origin sender, providing the authentication mechanism. [6] labels this as a "time-bound" feature because each share of the original secret has an associated timeframe in which the share is viewed once the period of time has been reached. Due to the continuous authentication features used in the protocol and its time related processes; all nodes are required to have time-synchronization implemented to allow these time-related processes to occur. This requirement is also described in [4] as a result of the incorporation of timestamps within the key management phases. The protocol allows the sender to transmit a message to many recipients, whilst enabling these recipients to authenticate the transmitted message and its origin source, providing protection for both the end-nodes and sender. The sender can transmit a challenge to each recipient end-node, protecting against masquerade attacks. To construct the challenge a chosen nonce by the recipient alongside a secret share that the sender uniquely designated to that recipient is hashed. Using the

authentication key, a MAC of the message for each r is produced. The sender creates a multicast message consisting of the multicast message and the MACs of the recipients, dispatching it to the recipients. Once received, each recipient then authenticates the sender and the message's integrity.

2.3 Key Findings

Analysis of the proposed protocols differed in methodology and produced varying results from concrete findings backed by statistics to theoretical findings based on evaluation metrics. Yao, et al. [3], and Bamasag and Toumi [6] took a more conceptual analysis approach with [3] evaluating performance based on seven aspects from the research paper "Seven cardinal properties of sensor network broadcast authentication [14]," and [6] choosing several similar aspects for evaluation. Carlier, et al. [4] and Park [5] took a more practical analysis approach with both implementing and simulating the protocols in ContikiOS and the Cooja simulator. [4] simulated multicast and unicast variations of the scheme measuring efficiency based on delay, packet fragmentation, energy consumption, and storage requirements. [5] simulated the old DTLS handshake and revised security bootstrapped variations of CoAP multicast measuring efficiency based on delay and packet loss. The findings of all four papers will be analysed based on the extent of the applicability, performance, and security delivered by the schemes. Analysis will reference a mixture of the cardinal properties from [3][14], evaluation metrics from [6], and resulting data from simulations in [4] and [5].

The scheme proposed by Yao, et al. [3] was discovered to work exceptionally well for small scale resource-constrained applications that are not suited to using signatures based on public key algorithms but had little to no application for large-scale multicast due to the high amount of communication being too resource-costly. On the other hand, Carlier, et al. [4], Park [5], and Bamasag and Toumi [6] all pride themselves on scalability since additional nodes can be included without major modifications to the rest of the protocol. The symmetric key scheme proposed by Carlier, et al. [4] was specifically designed for WSNs and through simulation was proven to be highly efficient, with multicast producing a 200ms reduction in delay in comparison to unicast with 5 members at 4 hops. Park [5] secured multicast for CoAP IoT applications and discovered the proposed security bootstrapping approach was significantly more efficient than the traditional DTLS Handshake model, with a roundtrip delay reduction of 28,000 ms for 5 servers at 2 hops. Comparatively, for 5 nodes/servers at 2 hops [4] is more efficient than [5] with a delay of 110 ms compared to 6000 ms. On one hand, this shows the effectiveness and benefits of using a symmetric key approach compared to a public key approach. Furthermore, other factors such as the reduction in network traffic achieved by using a publish/subscribe architecture as opposed to a request/response architecture, and the superior performance of the BMRF multicast protocol contribute to [4]'s better performance. On the other hand, one can argue the application of both protocols is different so comparing performance serves no purpose as both have been shown to improve performance in their own environments. Bamasag and Toumi [6] report superior efficiency in comparison to authentication protocols like [5] that use asymmetric operations, e.g. Elliptic Curve, as a result of the low computational and commutative overhead of symmetric hashing and MAC operations. This view is held by Yao, et al. [3] that found the protocol satisfied the cardinal properties of "Low Computation Overhead" and "Low Communication Overhead" in resource-

constrained environments. In theory, this suggests that like [4], the protocols proposed by [3] and [6] have the potential to perform better than [5] for IoT applications. However, the weakness of [3] for large-scale multicast and the lack of statistical evidence backing these claims mean both [3] and [6] are yet to prove their protocols' performance.

All four protocols were found to enhance security aspects and maintain confidentiality, integrity, and authenticity to some degree. Confidentiality is realised through secret shared common keys in [3], [4], and [6], and security associations with encrypted information in [5]. Yao, et al. [3] satisfied the cardinal property of "Resistance Against Node Compromise" due to secret shared keys between each source and receiver node meaning one compromised node does not affect the communication between others. However, an issue is discovered as communication between the compromised node and the other nodes which share information becomes insecure. This issue is identified in [5] as a compromised CoAP server means the secret information inside it can be accessed and used to mount attacks against the other group members. However, similarly to [4] it is addressed by the assumption of an intrusion detection mechanism being present and fixed by the GC (in [5]) or NMM (in [4]). Overall, Bamasag and Toumi [6] deal with compromised nodes in the strongest way by allowing unique challenges to be generated that must be solved for verification in every communication, preventing masquerade attacks. In the other protocols resistance to attack is assumed by the strength of the group managers (GC/NMM) and the cryptographic security of the protocols preventing initial intrusion. Carlier, et al. [4] proved that integrity of data and authenticity of communication is continuously checked and maintained through cryptographically secure hash functions. Similarly, Yao, et al. [3] and Bamasag and Toumi [6] used MACs to ensure data is tamper resistant and Park [5] used certificates and signatures. Moreover, Carlier, et al. [4] and Bamasag and Toumi [6] incorporated time-based approaches to ensure communication has a constrained integrity period and "Freshness", protecting against potential replay attacks. This achieves the "Continuous authentication" motivation behind this project as the received multicast message must always come from the original sender within a specific timeframe. Park [5] protected against replay attacks and ensures "Freshness" with an alternative method using a nonce value. Yao, et al. [3] labelled the lack of time constraints and less restricting rules as a feature achieving the cardinal properties "Messages Sent at Irregular Times" and "High Message Entropy." Yao, et al. [3] feel the authentication schemes lack-of dependence on time allows multicast communication to occur more frequently and less restricting rules on the multicast data enable message authentication with high entropy. Although this may be true it is clear in the case of Carlier, et al. [4] that high performance is possible with time restrictions and replay attacks pose a risk to [3].

Based on the key findings the reporter feels the motivation behind this project was best met by Carlier, et al. [4] and Bamasag and Toumi [6]. This was because of the discovered limitations of [3] in applicability and achieving "Continuous authentication", and the lower performance and lack of a "secret sharing scheme" in [5]. In the next section this report will discuss each paper's achievements in terms of contribution to the scientific community and possible enhancements will be suggested.

3 Achievements and Enhancements

The scheme proposed by Yao, et al. [3] achieved its original aims and produced a lightweight multicast authentication mechanism for resource constrained IoT devices through use of Nyberg's fast one-way accumulator. The security and performance of the authentication mechanism was both well implemented and documented by the writers. The scheme successfully fulfilled the 7 cardinal properties by Luk, et al. [14] that performance analysis was based on. [3] contributed a multicast authentication mechanism that is resistant against node compromise, has low computational and communication overhead, robustness against packet loss (in regard to protection against lost packets being used to forge signatures), ability to perform immediate authentication, suitability for multicast communication at any timings and message authentication with high entropy. Its contribution was novel in demonstrating how Nyberg's fast one-way accumulator could be adapted and secured for multicast authentication, and [3] can be considered an effective authentication mechanism for small scale IoT devices. The main weakness of [3] resides in its scalability being limited to small scale resource constrained IoT applications, encountering major issues as the scale increases. Furthermore, the security of some communication may be compromised if an authenticated node is hijacked or a replay attack occurs. Finally, although [3] offers protection against lost packets being used to forge signatures it still shows susceptibility to packets being lost in the first place, leading to multicast messages not being fully created.

[3] shows room for improvement as enhancements can be made to address the weaknesses in design and lack of experimentation. For [3] to have application in industry the algorithm should be altered to be compatible with larger scale multicast applications. In its current state [3] serves as a useful research piece but no implementation or simulation has been conducted to accurately measure its performance. In the future the researchers could consider how to implement the authentication mechanism on a multicast engine and accurately test the performance. Furthermore, it is necessary to include a mechanism for locating and separating compromised nodes from the network. Some additional security service or a group manager (as in [4] and [5]) should be considered that runs on each node is able to detect malicious activity to warn other nodes that communication is not secure. Either a time-based component or a nonce should be included to prevent replay attacks and ensure continuous authentication. Finally, the scheme should attempt to decrease packet loss through inclusion of a handshake protocol or similar method to increase the ability to receive the lost packets.

The scheme proposed Carlier, et al. [4] successfully met its aims, producing an efficient key management scheme for multicast authentication in WSNs with the minimum features of confidentiality, integrity, and authentication. Confidentiality was enforced throughout using secret keys, secure communication channels, and encryption/decryption with hash functions. Integrity was maintained by ensuring all information was cryptographically secured by hash functions, tamper resistant, and communication was kept within a constrained time frame. Authentication was continuous, checking the authenticity of all parties and communication throughout. Carlier, et al. [4] demonstrated that the scheme can run efficiently in a resource constrained environment, therefore proving the scheme's significance and application for WSNs in industry. The motivation behind this report was addressed as [4] demonstrated a continuous multicast authentication protocol with

a cryptographically secure secret sharing scheme. Little weakness can be found in [4] and the contribution was highly effective in showing both scientific and industry application. The only weakness identified by the reporter is [4] fails to discuss any mechanisms to detect abnormal behaviour and protect the trusted third party and NMM. It is assumed that these mechanisms are present but without proper analysis and simulation of attacks there could be security vulnerabilities present that were not considered. This research can therefore be enhanced by explaining the protection mechanisms in detail and expanding experimentation to simulate an array of attacks, proving the protocol's cryptographic strength.

The scheme proposed by Park [5] enhanced the security and performance of multicast communication with the constrained application protocol (CoAP) by replacing the DTLS Handshake with a security bootstrapping. [5] successfully achieved the aims of integrity and confidentiality of CoAP messages and improved reliability of multicast CoAP. Several novel advancements to CoAP multicast not seen in previous research were delivered such as protection of end-to-end communication using a session key between CoAP endpoints, a cryptographically secure ECQV implicit certificate issuance [11] based registration protocol, encryption of communication even over unreliable UDP, dynamic and scalable access control, and confinement of a compromised CoAP server to protect other nodes. The significant reduction in delay shown by simulation of the proposed security bootstrapped authentication mechanism proved the paper's significance of contribution in improving both CoAP multicast research and performance in industry. Much like Carlier et, al. [4], Park [5] showed real-world application and functionality in controlling IoT devices over the internet, but unlike the others a RESTful approach was shown due to the nature of CoAP.

Comparatively [5] was significantly less efficient than [4] and produced a greater delay. This weakness is likely a result of using asymmetric instead of symmetric cryptography, the multicast engine, and client/server instead of publish/subscribe architecture. Another weakness is the use of a resource directory (RD) server to store CoAP server information creates a potential single point of failure that could be used to attack the entire network if compromised. Furthermore, although Park [5] claims to confine the spread of a compromised CoAP server, attacks can still be mounted in the window of time until it is reported to the group controller. In this aspect [5] fails to satisfy the "Continuous authentication" motivation of this report. Park [5] also fails to address how the security channel between the group controller and clients/servers is secured, assuming it is secured by TLS/DTLS but stating that the security mode is "beyond the scope of this article." This is a major security risk that should have been addressed as an unsecured channel would allow keys and certificates to be intercepted and used by an attacker. Although [5] demonstrates an approach that shows clear superiority to previous CoAP multicast protocols there is some room for improvement. Possible enhancements to the scheme can be made by considering how symmetric operations and alternative multicast engines can replace asymmetric operations and inefficient routing protocols to accelerate performance. Security of [5] is stronger than traditional CoAP multicast but could be enhanced further by limiting the information stored on the RD server (where possible), ensuring security mechanisms are put in place to prevent compromise of nodes and effective intrusion detection, and extensive penetration testing of the system as no vulnerability testing was reported. Furthermore, Park [5] should extend the research to secure the channel between the group controller and clients/servers, ensuring all communication is encrypted

and cannot be intercepted by an attacker.

The scheme proposed by Bamasag and Toumi [6] contributed an efficient continuous multicast authentication protocol for IoT with a secret sharing scheme. [6] achieved all initial objectives and satisfied this project's motivation, delivering multicast authentication in an efficient, secure, scalable and timely way. The contribution of [6] has potential for wider use in multicast applications for both small-scale resource constraint environments and larger scale network. Similarly to [3], the algorithm used in this protocol is complex but easily computable by a computer. [6] is an extension of the researcher's previous work, providing continuous authentication utilising Shamir's secret sharing scheme [13] and protection against several potential threats identified in the threat model, with the protocols entire design based on securing them. As a result of this [6] can be deemed the most well secured authentication mechanism in comparison to the explored research. Furthermore, the security and potential performance has been well evaluated, documenting the 7 mains aspects the protocol fulfils. In comparison to the other research this protocol utilises the most time-based processes and relies on all the nodes (both sender and recipients) having time synchronization enabled. [6] does not suffer the same weakness as [3], being able to scale well outside of small-scale applications. The contribution of Bamasag and Toumi [6] is mainly scientific but with some extension could be used as a multicast authentication mechanism in industry. Little weakness can be found in the design of [6] as security has been carefully considered throughout and all functionality is performance focused for delivering maximum efficiency. However, a lack of evidence and simulation limits the contributions real-world applicability. Theoretically [6] could be the strongest multicast authentication mechanism but until it has been simulated a conclusion cannot be reached. Bamasag and Toumi [6] can enhance the research and its significance of contribution by prototyping the authentication mechanism. Only once the mechanism has been tested for varied numbers of nodes at different hops can it be deemed superior to [4]. Through simulation it may be discovered that the protocol is inefficient at certain distances, suffers from unexpected packet loss, or contains an unexpected vulnerability that must be addressed.

4 Conclusion

This report set out to find a secure and efficient continuous multicast authentication protocol for IoT applications implementing a secret sharing scheme. Four protocols were selected and analysed in the hopes of satisfying this criteria and comparative strengths/weaknesses were discovered. The revised fast one-way accumulator proposed by Yao, et al. [3] was found to be lightweight, well designed and cryptographically secure, performing effectively for small scale IoT devices. However, its poor scalability, missing continuous authentication property, and lack of implementation meant it offered no viability as a security mechanism in a real multicast system. The security bootstrapped extension of CoAP multicast communication proposed by Park [5] offered a significant improvement to CoAP multicast performance and security, demonstrating viable use in industry as a clear enhancement to current CoAP systems. However, in terms of the scope of this report its performance was less efficient than [4], several security issues were still present, and continuous authentication was not fully satisfied. The continuous multicast authentication protocol based on a secret sharing scheme proposed by Bamasag and Toumi [6] directly satisfied the motivation behind this report and theoretically offered the strongest multicast protection. However, despite

its potential the lack of documented implementation and testing created concern that its security and performance will be ineffective in reality. Overall, the reporter feels the symmetric key management protocol for multicast authentication in WSNs proposed by Carlier, et al. [4] was the strongest approach due to the full delivery of confidentiality, integrity, and authentication to multicast communication, and the extensive testing and documentation providing clear evidence of the protocols superior performance. In future, the reporter feels simulation of these four protocols working in lightweight IoT environments with similar conditions is necessary to accurately establish the protocol with the best performance and security.

5 References

[1] Statista, “Number of internet of things (IoT) connected devices worldwide in 2018, 2025 and 2030,” statista.com, 2019, [Online], Available: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/, [Accessed Apr. 20, 2020].

[2] Upasana, “Real World IoT Applications in Different Domains,” edureka.co, May. 22, 2019, [Online], Available: https://www.edureka.co/blog/iot-applications/, [Accessed Apr. 20, 2020].

[3] X. Yao, X. Han, X. Du, and X. Zhou, “A Lightweight Multicast Authentication Mechanism for Small Scale IoT Applications,” in Sensors Journal, IEEE, 2013, vol. 13, no. 10, pp. 3693-3701, doi: 10.1109/JSEN.2013.2266116.

[4] M. Carlier, K. Steenhaut, A. Braeken, “Symmetric-Key-Based Security for Multicast Communication in Wireless Sensor Networks,” in Computers, vol. 8, no. 1, p. 27, Mar. 2019.

[5] C. Park, “Security Architecture for Secure Multicast CoAP Applications,” in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3441-3452, Apr. 2020.

[6] O. Bamasag and K. Y. Toumi, "Efficient multicast authentication in internet of things," in 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2016, pp. 429-435.

[7] K. Nyberg, “Fast accumulated hashing,” in Gollmann D. (eds) Fast Software Encryption. FSE 1996. Lecture Notes in Computer Science, 1996, vol. 1039, Springer, Berlin, Heidelberg, doi: https://doi.org/10.1007/3-540-60865-6-45.

[8] S. Akkermans, R. Bachiller, N. Matthys, W. Joosen, D. Hughes and M. Vučinić, "Towards efficient publish-subscribe middleware in the IoT with IPv6 multicast," in 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, 2016, pp. 1-6.

[9] G. Gastón Lorente, B. Lemmens, M. Carlier, A. Braeken and K. Steenhaut, “BMRF: Bidirectional Multicast RPL Forwarding,” in Ad Hoc Networks, 2017, vol. 54, pp. 69–84.

[10] P. Triantafillou and A. Economides, "Subscription summarization: a new paradigm for efficient publish/subscribe systems," in 24th International Conference on Distributed Computing Systems, 2004. Proceedings., Tokyo, Japan, 2004, pp. 562-571.

[11] D. R. L. Brown, R. Gallant, and S. A. Vanstone, “Provably secure implicit certificate schemes,” in Financial Cryptography (LNCS 2339). Heidelberg, Germany: Springer-Verlag, Feb. 2001, pp. 156–165.

[12] O. Bamasag and K. Youcef-Toumi, "Towards Continuous Authentication in Internet of Things Based on Secret Sharing Scheme", in Proceedings of the WESS’15: Workshop on Embedded Systems Security, Amsterdam, The Netherland, 4-9 October 2015.

[13] A. Shamir, "How to Share a Secret", in Communications of the ACM, vol. 22, no. 11, 1979, pp. 612-613.

[14] M. Luk, A. Perrig, and B. Whillock, “Seven cardinal properties of sensor network broadcast authentication,” in Proc. 4th ACM Workshop Security Ad Hoc Sensor Netw., 2006, pp. 147–156.