Elliptic Curve Based Cybersecurity Schemes for Publish-Subscribe Internet of Things

Abebe Abeshu Diro(✉), Naveen Chilamkurti, and Prakash Veeraraghavan

La Trobe University, Department of Computer Science and IT, Bundoora VIC, Melbourne 3086, Australia
{a.diro, n.chilamkurti, P.Veera}@latrobe.edu.au

Abstract. The rapid increase in the number of connected things across the globe has been brought about by the deployment of the Internet of things (IoTs) at home, in organizations and industries. The innovation of smart things has been envisioned through various protocols, but the most prevalent protocols are publish-subscribe protocols such as Message Queue Telemetry Transport (MQTT) and Advanced Message Queuing Protocol (AMQP). One of the major concerns in the adoption of such protocols for the IoTs is the lack of security mechanisms as the existing security protocols cannot be adapted due to their large overhead of computations, storage and communications. To address this issue, we propose a lightweight protocol using Elliptic Curve Cryptography (ECC) for IoT security. We present analytical and simulation results, and compare the results to the existing protocols of traditional Internet.

Keywords: Cyber security · Publish-subscribe systems · Internet of things · Elliptic curve cryptography

1 Introduction

By 2020, the number of connected things will be more than 6 times of the world population as operational technologies such as those in the factory and home are becoming the part of connected entities coupled with Information technology entities that are currently in use for daily purposes [1]. It means that there are more than six smart things for every person on the globe. This evolutionary paradigm is brought about by the emergence of smart things capable of collecting, processing and communicating data among themselves or interacting humans pervasively. Though IoT has several promises and potentials, its deployment might pose various security issues due to their unattended nature and their limited resources. Traditional cryptography systems such as RSA have been used as security solutions on the Internet [2–5], but they are not practical to implement for IoT devices due to their overheads in computations, storage and communications of security parameters such as keys. For instance, RSA assumes that the increment of the key size increases the level of security if the key itself is not unveiled, and consequently, the overheads incur high the consumption of resources. For this reason, a more efficient public key cryptographic mechanisms are required. To address this limitation, elliptic curve based cryptographic scheme could be applied in the existing

J.-H. Lee and S. Pack (Eds.): QShine 2016, LNICST 199, pp. 258–268, 2017.
DOI: 10.1007/978-3-319-60717-7_26

pub-sub communication popular protocols of IoTs such as MQTT [6]. MQTT was designed in many-to-many communication protocol paradigm for disseminating messages between subscribers through a central entity in the emerging IoT applications such as social networks, V2V, WSNs. A crucial characteristic of these protocols as a pub-sub system are the decoupling of publishers and subscribers, enabling a many-to-many communication model. Such a system presents many benefits as well as potential security risks regarding authenticity, confidentiality, integrity and availability. Unfortunately, most of the existing researches on pub-sub networks focus only on performance and scalability. Very few papers have devoted to developing a novel security framework that can resist multiple security problems inherent in them. While RSA is a well-established protocol for Internet communications, it is not lightweight to be proposed for resource limited IoT environments due to its dependence on resource intensive public key cryptography. Hence, this paper deals with lightweight cyber security issues for publish/subscribe mode of communication. The contributions of this paper are:

- We propose novel lightweight security solutions for publish-subscribe protocol based Internet of Things using ECC. Compared to RSA protocols, our scheme could provide the same level of security for publications and subscriptions while it decreases computation, communication and storage costs.
- Scalable key exchange mechanism with less number of handshakes in linear time compared to RSA.

2 Public Key Cryptography

Cryptography is defined as the mechanism of secure communication in which designing and analyzing of protocols that can combat cyber-attacks is crucial. Public key cryptographic systems have become the modern way of cybersecurity revolution over an insecure communication channel. Some public-key schemes are discussed in the following section.

2.1 RSA

RSA is an acronym for Rivest, Shamir and Adleman after its inventors back in 1977 as a public key cryptographic system. The security of RSA lies on the computational difficulty of factorization of large prime numbers [7]. The authors, in their study, suggested that the method could be for encryption and digital signature. In the encryption/decryption process, a public key (e, n) and a private key (d, n) are used where all the parameters are positive integers. The encryption and decryption process are shown as follows:

C = E(M) = Me(mod n), where M = message
D(C) = Cd(mod n), where C = cipher text
*n = product of two large prime numbers p and q (n = p * q)*
*d = large random relative prime to p (i.e. gcd(d,(p − 1) * (q − 1)) = 1)*
*e = multiplicative inverse of d modulo (p − 1) * (q − 1).*

2.2 Diffie-Hellman Key Exchange Protocol

Diffie-Hellman key exchange protocol was proposed in 1976 by Whitfield Diffie and Martin E [4]. Hellman as key distribution scheme over insecure media as opposed to other cryptographic systems which need secure channel for the distribution. The algorithm depends on the difficulty of solving discrete logarithmic problem. In the key exchange process, partner A randomly chooses secret key xA from the interval of [0, q] to calculate yA = α xA Mod(q) for sending publically to B, and partner B selects secret key xB from the same interval to compute yB = α xB Mod(q) to send publically to A. Partners A and B establish shared keys (K_{AB} = α xA xB Mod(q)) using the combination of their secret keys over insecure channel. Even if the adversary gets one of the secret keys and computes K_{AB}, it is difficult to solve the discrete algorithmic problem as it has no knowledge of xA and xB. One of the drawbacks of this algorithm is that it lacks the mechanism of authentication and suffer from man-in-the-middle of attack.

2.3 Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) was introduced at the same time by Victor S. Miller and Neal Koblitz in 1985. ECC can provide an analogy to the Discrete Logarithm (DL) based systems such as Diffie-Hellman in the algorithm known as Elliptic Curve Discrete Logarithm Problem (ECDLP) [4, 5]. ECDLP states that an elliptic curve E over $G_F(q)$ and two points P, Q ϵ E, compute an integer x such that Q = xP. An elliptic curve E is formulated by $y^2 = x^3 + ax + b$ where $4a^3 + 27b^2 \neq 0$ and a, b, x are elements of a finite field F. The point addition of E given two points $P(x_1, y_1)$ and $Q(x_2, y_2)$ on E, with P, Q $\neq \infty$ is $R(x_3, y_3) = P(x_1, y_1) + Q(x_2, y_2)$, and defined as follows:

- If $x_1 \neq x_2$ then $x_3 = m^2 - x_1 - x_2$, $y_3 = m(x_1 - x_3) - y_1$ *where* $m = (y2 - y1)/(x2 - x1)$
- If $x_1 = x_2$ but $y_1 \neq y_2$ then P + Q = ∞
- If P = Q and $y_1 \neq 0$, then $x_3 = m^2 - 2x_1$, $y_3 = m(x_1 - x_3) - y_1$ where $(3x_1^2 + a)/2y_1$
- If P = Q and $y_1 = 0$, then P + Q = ∞
- P + ∞ = ∞

ECC is an algebra based light-weight next generation cryptography, which provides the at least the same level of cybersecurity solutions with smaller key and message size compared to other public key cryptographic systems such as RSA. The following table gives the comparison of key size between ECC and RSA. It seems that ECC could yield a desired level of security with a key size of 256-bits that RSA scheme requires a key size of 3072-bits to achieve. ECC could be used for digital signature to verify the digital content and the source, integrated encryption to secure plain and cipher texts, key management (Diffie-Hellman) to share keys secretly over insecure channel. The slow adoption of ECC so far seems to change with the fast growth of IoT devices with limited resources to achieve a desired security level without compromising performance. Due to this, the ECC approach for cybersecurity is very appealing for small devices such as meters, smartphones and embedded devices as it reduces computational time, data transmitted and stored.

3 Related Work

Publish-subscribe networks could be direct channel and pub-sub network depending on the scheme used for disseminating information [8]. In a direct channel mechanism, a publisher directly passes a publication to subscribers under specific topic of subscription. However, in pub-sub system publishers and the subscribers communicate via an intermediate broker which facilitates the publication/subscription. This kind of model is more scalable than the previous in that publishers tend to become less performance bottle-necked. Most studies on pub-sub systems have concentrated on performance, scalability and availability [9]. Unfortunately, very limited studies are present about the security aspects of these systems in traditional Internet, and almost none for pub-sub systems in IoTs protocol such as MQTT. The considerable amount of research has been done in secure group communication fields [10]. The major problem with such systems is that group key management is not as flexible as pub-sub systems. Additionally, protocols such as RSA cannot be adapted to IoT environments due to their heavy weight nature [11]. Group key management in securely distributing of events of content-based pub-sub network has also been analyzed by Opyrchal et al. [12]. This kind of arrangement increases the number of keys exponentially as subscribers increase, and hence, suffers from scalability. In contrast, our system permits flexible joining to and leaving from the network without compromising security and performance. Wang et al. [13] analyze the security requirements in a content-based pub-sub system, identifying authentication of publications, integrity of publications, subscription integrity and service integrity as the key issues. The paper is detail enough in the context of general Internet, but fails to work for the Internet of things whose resource constraint is high. EventGuard [14] has shown the possibility of achieving security requirements, but it is not applicable for IoT devices because of its resource demand, content-based networking is not widely accepted yet.

4 System Design

In pub-sub communication scheme, broker plays vital role in handling subscriptions, publications and information disseminating under a specific topic. In our design, end nodes communicate securely with a broker in pub-sub paradigm under their respective subscription. The broker is assumed to have a considerable computational and storage power for key generation and management per session for all subscribers. The system enforces integrity and access control of messages under a given topic by employing authorization and encryption key for publishers and decryption keys for subscribers (Fig. 1).

4.1 System Goals

Our security protocol design has basically three sets of design goals: security, performance and scalability goals. In pub-sub system, publishers/subscribers should be authentic to broker and vice versa to avoid impersonated publications/subscriptions.

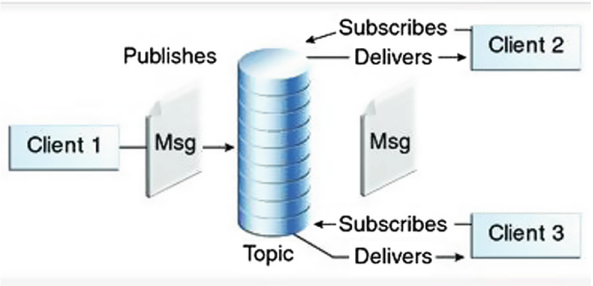

Fig. 1. General architecture of topic based pub-sub system security

This is to prevent unauthorized subscribers from accessing the topics for which they have not subscribed. In addition, non-publisher entity should not create a message for which a subscriber claims subscription. In the case that many publishers write on common topic, subscribers should be able to authenticate the actual publisher. It is required that messages sent from publisher to subscriber via broker is guarded against disclosure or modification. These includes authorized publications/subscriptions, subscription privacy and routing integrity. The security framework should also be resilient against Denial of Service (DoS) attacks such as flooding attacks, fake unsubscribe and selective or random dropping attacks. The system is expected to scale with the number of publishers and subscribers in the network. The security mechanism should not add performance overhead to the existing pub-sub system.

4.2 Security Procedures

Parameter Settings
Input: Elliptic curve E_p (a, b): $y^2 = x^3 + ax + b$ (modp) over subgroup (finite field) Zp where p is large prime number, a, b ϵ Zp ($4a^3 + 27b^2$(modp) $\neq 0$)
Output: secret master key K_m, public key K_p, Fog broker (subscription manager) key (K_s), subscriber private key (K_i)

1. Choose generator G from an elliptic curve point over
2. Choose random master secret key K_m from Z_p and computes public key $H = GK_m$
3. Broadcast public parameters $K_p \leftarrow (E_p, G, p, H)$
4. Choose random subscriber key K_i from Z_p and calculate Fog side key $K_s \leftarrow K_m \oplus K_i$
5. send K_i to subscriber and K_s to Subscription manager.

Subscription
Input: subscriber identity (ID), topic, Fog broker (subscription manager) key (K_s), subscriber private key (K_i)
Output: authentication key K

1. Subscriber requests by presenting ($topic_i$, r, ID_i)
2. Subscription Manager sends autho: ($topic_i$, ID_s, $rGHK_sN$, uSub = H ($topic_i \| ID_i$) to subscriber

3. Subscriber computes, $rK_i \oplus rGHKsN = HGK_mN = (GK_m)\ GK_mN = N$ and sends ({{r′, r′ GHK_iL}} where L = (N – 1, K))
4. Subscription Manager computes, r′Ks $\oplus$ $rGHK_iL = HGK_mL = (GK_m)\ GK_mL$ = L = N – 1. Here, mutual authentication is valid, and K is sent to Fog broker (topic manager)
5. Subscription Manager sends H ($topic_i$, ID_i, K) to publication manager to store to be used in procedure 3.

Publication
Input: Input: subscriber identity (ID), topic, authentication key K, message M, subscriber private key (K_i), subscription manager key (K_s)
Output: intermediate cipher texts, plain text

1. Publisher requests publication by H ($topic_i$, ID_i, K)
2. Publication manager compares H (topic*, ID * i, K*) ?= H (topic, Idi, K) and sends (IDs, H ($topic_i$, ID_i, K))
3. Publisher sends ($topic_i$, ID_i, {r_i, C = r_iGK_i M}) (encryption of message M)
4. Publication manager recalculates Cs = $rGK_s \oplus C = rGK_s \oplus rGK_i$ M. = K_m.M. Then $C_i = C_s \oplus rGKs = rGK_i$.M, and sends C_i to the subscriber
5. Subscribers decrypt $C_i = rGK_i$.M by calculating rGK_i.M $\oplus$ rGK_i = M.

Unsubscription

1. The subscriber that needs to leave the group sends topic, IDi, K, uSubI(topic) to the cloud broker
2. Broker checks the unsubsciber by computing (topic*, D * i) ?= topic, Di and informs the key generator for key revocation
3. Key generator module unsubscribes the subscriber by send acknowledgement.

5 Analysis

The threat model is composed of subscription, publication and broker mediation processes. It is assumed that the key generator is secured, and trusted to provide keys for all the operations of end devices. Pub-sub networks, like access control schemes, need entities to get read/write access before performing the appropriate actions such as read action for subscribers, and write action for publishers. The devised protocol could be evaluated in terms of various overhead, scalability and security parameters.

5.1 Performance Analysis

In our algorithm, much of computational and storage overheads were offloaded to the broker which is richer in resource than publisher/subscriber IoTs devices. We employed less expensive computations such as XORing and elliptic curve point additions using ECC 160-bit curve (secp160rl) in which EC point is 20-bytes. The superiority of our scheme in resources (storage, computations, and communications) conservation, compared to RSA, could be seen from Table 1. The system also saves

Table 1. key size comparison of ECC and RSA

ECC key size	RSA key size	Ratio
160 bits	1024 bit	1:6
224 bit	2048 bit	1:9
256 bit	3072 bit	1:12
384 bit	7680 bit	1:20
512 bit	15360 bit	1:30

much of communication bandwidth in reducing the number of handshakes compared to the already existing heavyweight protocols such as RSA. For instance, our first scheme incurs only a total of 218 bytes of storage overhead for both publisher/subscriber and Fog server during subscription, while RSA systems might occupy over 6440 bytes for similar settings in a single connection. During publication, a single connection consumes 198 bytes of storage space in our protocol, and the existing protocol consumes over 6440 bytes. The case of communication burden could also be explained in a similar manner for both subscription and publication as it can be seen from the table. The second scheme is even more efficient than the first scheme in offloading storage and communication burden from publisher/subscribers, but it is slightly more expensive computationally. However, the burden on the broker is comparable in the both protocols, which is less than the overhead of RSA. Thus, our system is more efficient in terms of delay, storage and computation than RSA systems for pub-sub based IoT connections in Fog computing.

5.2 Scalability Analysis

The scalability issue, which is the most important factor for secure pub-sub systems, could be seen in terms of key exchange when a node joins or leaves a network. The broker handles publication and subscription with very small number of handshakes, and it does not need to update subscribers' keys frequently. In addition the key sizes and run times scale linearly for ECC with increasing security level while for RSA they scale super-linearly. On the other hand, it is difficult to manage the keys in subscriber group systems as it needs processing cost of O(2n) for managing keys for n subscribers, while our scheme needs at most logarithm of the number of topics (Fig. 2 and Table 2).

5.3 Security Analysis

This section evaluates the basic security of the proposed system. It is logical to begin with preliminary concepts required to understand the analysis and proof, and then show the security of subscriptions and publications. Basically a mechanism is said to be secured the adversary's advantage in breaking the scheme is a negligible function of the security parameter.

Theorem 1: (Negligible Function). A function f is negligible if for each polynomial p() there exists N such that for all integers $n > N$ it holds that $f(n) < 1\ p(n)$. Assuming that

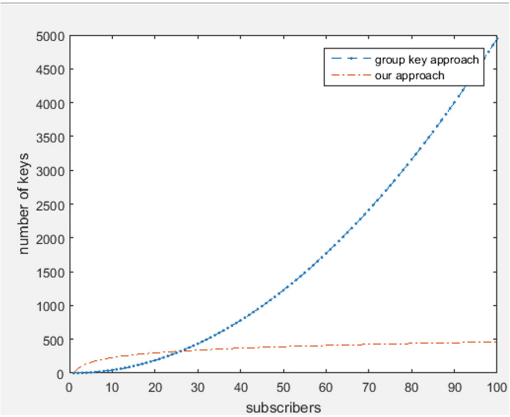

Fig. 2. Comparison of the number of keys

Table 2. Performance comparison between our scheme and RSA

	Parameter type	Our scheme		RSA	
		Publisher/subscriber	Fog broker	Publisher/subscriber	Fog broker
Subscription	Computational overhead	-/1PM,2XOR	2PM, 2XOR		
	Storage overhead	-/86 bytes	132 bytes	-/Over 340 bytes	Over 6100 bytes
	Communication overhead	-/66 bytes	132 bytes	-/Over 340 bytes	Over 6100 bytes
Publication	Computational overhead	1PM, 1XOR	1C, 2PM,2XOR		
	Storage overhead	-/66 bytes	132 bytes	-/Over 340 bytes	Over 6100 bytes
	Communication overhead	34 bytes	28 bytes	-/Over 340 bytes	Over 6100 bytes

the adversary is with bounded resources in PPT, the scheme should be secure and the success probability of any such adversary is negligible. Our protocol depends on a pseudorandom function f whose cannot be distinguished by adversary.

Theorem 2: (Pseudorandom Function). A function f: $\{0, 1\} * \times \{0, 1\} * \rightarrow \{0, 1\} *$ is pseudorandom if for all PPT adversaries A, there exists a negligible function negl such that: $|\Pr[Afk(\cdot) = 1] - \Pr[AF(\cdot) = 1]| < \text{negl}(n)$ where $k \rightarrow \{0, 1\}$ n is chosen uniformly randomly and F is a function chosen uniformly randomly from the set of functions mapping n-bit strings to n-bit strings. Our proof depends on the assumption that the Elliptic Curve Diffie-Hellman (ECDH) is hard in a group G, i.e., it is hard for an adversary to distinguish between group elements $\alpha\beta P$ and γP given αP and βP.

Theorem 3: (ECDH Assumption). The Elliptic Curve Diffie-Hellman (ECDH) problem is hard regarding a group G if for all PPT adversaries A, there exists a negligible

function negl such that |Pr[A(G, q, P, αP, βP, αβP) = 1] – Pr [A(G, q, P, αP, βP, γP) = 1]| < negl(k) where G is a cyclic group of order q (|q| = k) and P is a generator of G, and α, β, γ ∈ Zq are uniformly randomly chosen. The schemes we are using in our solution is proven to be indistinguishable under chosen plaintext attack (IND-CPA) and we will prove that our scheme is also IND-CPA secure. A cryptosystem is considered IND-CPA secure if no PPT adversary, given an encryption of a message randomly chosen from two plaintext messages chosen by the adversary, can identify which message was encrypted with non-negligible probability.

Theorem 4: If the ECDH problem is hard relative to G, then our EC based scheme is indistinguishable under chosen plaintext attack (IND-CPA). That is, for all PPT adversaries A there exists a negligible function negl such that

SuccessA, p(k) = pr[b′ = b|(pk, Km) ← Init(1k), (Ki, Ks) ← GenKey (Km, Di), m0, m1 ← AEnc(Ki,.)(Ks) b ← R{0,1}, ci * (mb) = Enc(Ki, mb), b′ ← AEnc(Ki,.)(Ks, ci*(mb))] < $\frac{1}{2}$ + negl(k).

Proof: Assuming PPT adversary A′ attempting to solve the ECDH problem using A function and having G, q, P, αP, βP, αβP or γP as input for some random α, β, γ, A′ performs the following computations:

- Sends public parameters G, q, P to A, and then, by randomly choosing Ks ← R Zpfor each IDi. Then it computes KiP = αP ⊕ KsP. It sends all (IDi, Ks) to A and stores all (i, Ks, KiP).
- In order to access encryption algorithm, A passes m to A′, and A′ chooses randomly r ← Zq and replies with (rP, rPKiM)
- A produces m0, m1. A′ selects a random bit b and sends βP, βPKs ⊕ X mb to A, where X = γP or αβP
- A produces b′, and If b = b′, A′ outputs 1, otherwise 0.

Case 1: Since γ is randomly chosen, and hence γP, then βPKs ⊕ γPmb reveals no information about mb as it is a random element of G i.e. uniform distribution irrespective of mb value. Adversary A must distinguish between m0 and m1 without additional information. The success probability of b′ = b is exactly 1/2 when b is chosen uniformly randomly, and A′ outputs 1 iff A outputs b′ = b, in which case Pr[A′(G, q, P, αP, βP, γP) = 1] = 1/2.

Case 2: In the case parameters X = αβP, and βPKs ⊕ αβP mb = βP(Ks ⊕ α) = PKi, then βPKs ⊕ X mb is valid cipher. In this case, case Pr[A′(G, q, P, αP, βP, αβP) = 1] = SuccessA, p(k). Assuming ECDH is hard to break in group G, then

$$\begin{cases} |\Pr[A'(G,q,P,\alpha P,\beta P,\alpha\beta P)=1]-\Pr[A'(G,q,P,\\ \qquad \alpha P,\beta P,\gamma P)=1]|<\mathrm{negl}(k) \\ \Pr[A'(G,q,P,\alpha P,\beta P,\alpha\beta P)=1]<1/2+\mathrm{negl}(k) \end{cases}$$

$$\therefore \mathrm{SuccessA},\, p(k)<1/2+\mathrm{negl}(k)$$

Theorem 5: The proposed scheme could provide mutual authentication between publishers and the broker, and the subscribers and the broker, and hence resist man-in-the-middle attack.

Proof: Mutual authentication means that publishers and subscribers could authenticate with cloud broker during authentication. In the scheme, the key generator provides key Ki for clients and key Ks for broker in such a way that master key Km = Ki + Ks. The key pair, coupled with nonce and session key, is used to check the correctness of the identities of the interacting parties. By mutual authentication, the scheme provides resistance for man-in-the-middle attack. Authentication enables fine-grained subscribers and publishers access control mechanisms on top of the encryption scheme. If the broker is trusted, we can let the broker authenticate the users and enforce our authorization policies.

Theorem 6: The proposed architecture could provide known-key security.

Proof: Known key security means that unique session key is stablished between nodes and broker at the end of authentication scheme. The protocol generates unique session key K at every session ensuring that known-key attack is not possible.

Theorem 7: The scheme could withstand replay attack.

Proof: Replay attack means the impersonation of publishers/subscribers or the broker by the adversary by exploiting the previous session information. The adversary might request the cloud broker by sending (IDi, topic, ri), but since the random numbers and time stamps from both sides are generated every session replay attack is impossible.

Theorem 8: The protocol could resist DDoS attack of malicious publication.

Proof: The use of random numbers in the messages originated from publishers, subscribers and the broker prevents the parties from malicious flooding of the broker by fake subscriptions or publications or un-subscriptions.

6 Conclusion and Future Work

In this paper, we have analyzed the possible application of Elliptic Curve cryptography for securing IoTs in pub-sub communication model. This lightweight scheme provides better scalability, and less overheads such as storage, communication than RSA based schemes employed in SSL/TSL while it guarantees the same level of security. As part of future study, we need to implement the protocol on real IoT platform such as Arduino.

Acknowledgment. Our thanks to Pervasive Computing and Networking Lab, La Trobe University, Melbourne, Australia for material and financial support.

References

1. Securing the Internet of Things: A Proposed Framework (2016). http://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html
2. Shen, H., Kumar, N., He, D., Shen, J., Chilamkurti, N.: A security-enhanced authentication with key agreement scheme for wireless mobile communications using elliptic curve cryptosystem. J. Supercomput. **72**, 3588–3600 (2016)

3. Zhang, Z., Qi, Q., Kumar, N., Chilamkurti, N., Jeong, H.-Y.: A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. Multimed. Tools Appl. **74**(10), 3477–3488 (2014)
4. Hankerson, D., Vanstone, S., Menezes, A.J.: Guide to Elliptic Curve Cryptography. Springer, Heidelberg (2004)
5. Sandeep, S.: Elliptic curve cryptography for constrained devices. Ph.D. Dissertation (2006)
6. Singh, M., Rajan, M.A., Shivraj, V.L., Balamuralidhar, P.: Secure MQTT for Internet of Things (IoT). In: 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT), pp. 746–751, 4–6 April 2015
7. Mitchell, J.C., Shmatikov, V., Stern, U.: Finite-state analysis of SSL 3.0. In: Proceedings of the 7th Conference on USENIX Security Symposium (SSYM 1998), Berkeley, CA, USA, vol. 7, p. 16. USENIX Association (1998)
8. Fiege, L., Zeidler, A., Buchmann, A., Kilian-Kehr, R., Mühl, G., Darmstadt, T.: Security aspects in publish/subscribe systems. In: Third International Workshop on Distributed Event-based Systems (DEBS 2004) (2004)
9. Gupta, V., Wurm, M., Zhu, Y., Millard, M., Fung, S., Gura, N., Eberle, H., Shantz, S.C.: Sizzle: a standards-based end-to-end security architecture for the embedded internet. Technical report, Sun Microsystems, Inc., Mountain View, CA, USA (2005)
10. Porambage, P., Braeken, A., Schmitt, C., Gurtov, A., Ylianttila, M., Stiller, B.: Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for IoT applications. Access IEEE **3**, 1503–1511 (2015)
11. Srivatsa, M., Liu, L.: Secure event dissemination in publish-subscribe networks. In: Conference on Distributed Computing Systems (2007)
12. Opyrchal, L., Prakash,A., Agrawal, A.: Designing a publish-subscribe substrate for privacy/security in pervasive environments. In: Proceedings of the 2006 ACS/IEEE International Conference on Pervasive Services, pp. 313–316, 26–29 June 2006
13. Wang, C., Carzaniga, A., Evans, D., Wolf, A.L.: Security issues and requirements for Internet-scale publish-subscribe systems. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences, HICSS 2002, pp. 3940–3947 (2002)
14. Srivatsa, M., Liu, L., Iyengar, A.: Eventguard: a system architecture for securing publish-subscribe networks. ACM Trans. Comput. Syst. **29**, 4 (2011)