IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 8, 2018, accepted March 24, 2018, date of publication April 6, 2018, date of current version June 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2822822

Analysis of Lightweight Encryption Scheme for Fog-to-Things Communication

ABEBE ABESHU DIRO[1], NAVEEN CHILAMKURTI[1], AND YUNYOUNG NAM[2]
[1]Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC 3083, Australia
[2]Department of Computer Science and Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Yunyoung Nam (ynam@sch.ac.kr)

This work was supported in part by the Soonchunhyang University Research Fund and in part by the MSIP (Ministry of Science, ICT, and Future Planning), South Korea, through the Information Technology Research Center Support Program Supervised by the Institute for Information and Communications Technology Promotion under Grant IITP-2018-2014-1-00720.

ABSTRACT The growing concerns in cybersecurity is preventing unknowns which evolve from time to time. Internet of Things (IoT) is one of the emerging fields that have been applied for smart cities and industries. The promises of IoTs could be confronted with the growth in the number and sophistication of cyberattacks. The extension of digital world into physical environment adds new attack surfaces on the existing security threats of traditional Internet. The major challenge brought about by physical connectivity of IoTs is to implement distributed security mechanisms for resource constrain of IoT devices. As an emerging architecture supporting IoT applications, fog computing can be considered to solve the resource and distribution issues in securing fog-to-things communication. Security functions and services, such as cryptography, could be offloaded to fog nodes to reduce computational and storage burdens on IoT devices. The distribution of fog nodes can also solve the scalability of cloud by reducing central processing and communications. On the other hand, lightweight cryptographic functions, such as elliptic curve cryptography, have been proved to be suitable for embedded systems. In this paper, we have analyzed security challenges in terms of cybersecurity principles and proposed a novel encryption scheme for fog-to-things communication.

INDEX TERMS Cybersecurity, Internet of things, fog computing, elliptic curve cryptography, lightweight cryptography.

I. INTRODUCTION

By the end of this decade, the exponential growth in the number of connected smart things, known as IoTs, is estimated to be about 6 times the population of the world. The adoption speed of these smart devices is unprecedentedly about five folds of the adoption history of electricity and telephony altogether. The main contributing factor for this increment in connectivity is the digitalization of home devices (refrigerators, fans), smart city applications (connected cars, smart traffic lights, smart grids, smart water utilities) and operational technologies (factory machines) across the globe. The trend has been ignited by the shift from IPv4 based Information technology (IT) to IPv6 oriented operational technology (OT). The invention of IPv6 is one of the crucial enablers of the deployment of IoT as Ipv4 is unable to meet the requirements of the massive connections of IoT networks [1], [2].

The massive scale adoption of IoTs, and the big data generation in the vicinity trigger businesses and industries to rethink the architecture of data processing, storage and communications. The myriad of cloud computing applications in business, industries, public services, etc. have been magnificent over the last decades. Cloud computing has brought essential breakthroughs of seamless IT outsourcing capability with value-added services for customers. However, the current explosion of edge computing paradigms has challenged the scalability and performance of centralized cloud for IoT applications [3]. Real-time applications such as smart cities, eHealth, intelligent transport systems, industries, etc. need predictable and low latency, and distributed low bandwidth communication from IoT end to data repositories, where cloud computing cannot satisfy the requirements. To mitigate these issues, distributed intelligence, known as fog computing (FC), that bridges cloud computing closer to the things has been introduced. Fig.1 shows typical fog architecture in the things-to-cloud continuum.

FC complements cloud computing as it forms a service continuum between IoT and the cloud. Fog network mimics the cloud-to-things way of interaction at the edge network. It bridges the gap between the cloud and smart things, which enables a service continuum. The gap is closed in the form of enabling the distribution of computing and control, storage,

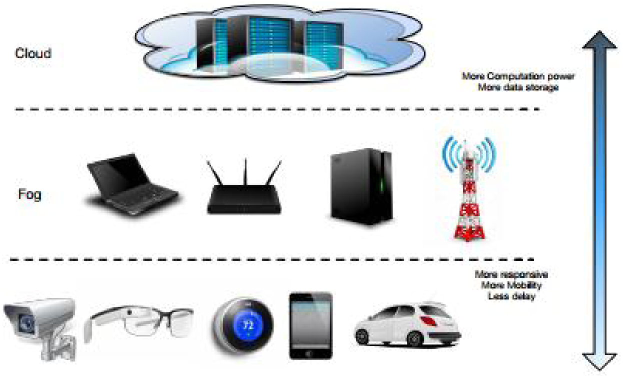

FIGURE 1. Basic Architecture of a fog network [8].

and networking functions closer to smart objects [4], [5] at any location across the continuum.

II. SECURITY SCHEMES IN FOG-TO-THINGS COMMUNICATION

The importance of security mechanisms for unprotected smart devices is unquestionable. For instance, smart grid users need authentication and authorization mechanisms to enable only subscribed users can access electric bills. Similarly, in smart life, healthcare systems require only physicians and nurses to notify about the status of patients as confidentiality has ultimate importance. In both cases, unauthorized adversaries should not eavesdrop on or access to the messages sent by clients.

In this section, we discuss the security challenges, threats, requirements, fog nodes based security architectures and the possible cryptographic solutions of fog-to-things communications.

A. SECURITY CHALLENGES

The promises of FC could be challenged by the growth in the number and sophistication of cyber-attacks in the communications of fog-to-things. The existing security threats of the traditional Internet will continue to be the threats of fog-to-things interactions. It is also evident that the extension of core networks to the physical world brings more devices, interactions, and protocols which can broaden attack surfaces and born new cyber threats. The major challenge brought about by physical connectivity is to implement distributed security mechanisms for fog-to-things communication for resource constrain of IoT devices [6]–[9]. It is bandwidth inefficient, prone to high latency and suffer from scalability to offload security functions of massively distributed IoT devices to cloud while it is computationally prohibitive to deploy security schemes on the devices. For instance, smart meter microcontroller has no capability of performing traditional Internet cryptographic operations, and the connection to cloud incurs a significant bandwidth cost and high latency for wireless communication dominant IoT environment. The emerging field of IoT needs a robust and lightweight security schemes. In traditional Internet, it is either the device or the centralized cloud that handles resource-intensive cybersecurity operations such as cryptographic encryption, access control, authentication, and authorization. These existing cybersecurity schemes for resource-rich infrastructure cannot prevail for addressing IoT cybersecurity challenges. The distribution and resource limitations of IoT devices in securing IoTs could be tackled by offloading securing functions to the distributed fog nodes [10]. Thus, fog nodes can be employed to offload cryptographic computations as proxy nodes without revealing the data in communications.

B. SECURITY THREATS

The IoT/Fog computing ecosystem is partly confronted with the same cybersecurity challenges as traditional IT ecosystem. IoT devices add a completely different dimension to cybersecurity world because of their physical interaction with the Internet. This is a serious implication that shows the transformation of attack surfaces from digital world (data) to physical world (actuation), which can broaden the horizon of known threats to zero-day attacks of new devices, workflows, and protocols [11]. The attack surface of fog-to-things can further be expanded as closed operational systems such as SCADA are moving to IP based open systems.

The major attacks in fog-to-things are impersonation, M-in-M, injection and DoS attacks [17]–[19]. Impersonation attack is a form of attack pretending to have legitimate identity using some other entity's identity. In the fog-to-things scenario, IoT devices could be eavesdropped or sniffed for identity. As wireless communication is the common platform of communication in the ecosystem, it is one of the most commonly observed attacks in the IoT use cases. This kind of spoofing attack might target the identity-based authentication systems which use MAC and IP addresses. Man-in-the-middle (MitM) attack is another type of impersonation attack in which the third node intercepts the communications between two nodes to capture data or authentication credentials. For instance, a simple mobile node (e.g. carrying Zeus) can act as a middleman to impersonate a sensor and its communicating fog node in the domain of Fog-to-things network. In addition, a replay attack occurs when an adversary captures some parts of a communication between two nodes and then retransmits the captured information, usually security credentials, later to bypass authentication mechanisms. A rogue fog node could be installed as a replay or MitM node in the Fog network either by the legitimate internal entity or cyber attacker without explicit authorization. For instance, a malicious rogue smart grid aggregator can tamper the data with smart meter creating wrong readings or it can modify IP addresses. Since it is stealthy to detect, a rogue node can instantiate further attacks such as DoS, causing a threat to data security and privacy [27], [28]. These indicate that strong identity management and mutual authentication, and encryption are the key elements for enhancing the security and privacy of IoT.

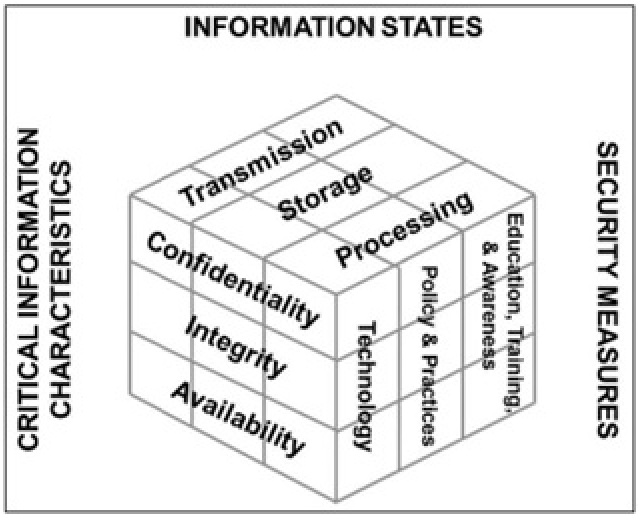

FIGURE 2. McCumber cube [12].

C. SECURITY REQUIREMENTS

The important design aspects of robust cybersecurity schemes stem from security goals, information states, and safeguards. These three general dimensions known as Cybersecurity Sorcery Cube (McCumber Cube) were created by [12] as a security framework for managing, evaluating and protecting systems and networks. Fig.2 shows the McCumber Cube. In this section, we focus on security goals.

The goals of cybersecurity known as the CIA triad are identified on the first dimension of cybersecurity cube and are widely used as a benchmark for evaluating and protecting cybersecurity. These goals, consisting of confidentiality, integrity, and availability [13] are the basic principles of the cybersecurity protection, and are the requirements for IoT security in the fog-to-things communications.

Confidentiality is the principle that prevents the disclosure of data or information to unauthorized users, resources, or processes. This is a critical requirement for fog-to-things interaction as the underlying wireless environment is less protected than wired network. Mechanisms of ensuring confidentiality, sometimes known as privacy, include cryptographic methods such as data encryption, authentication, and access control. Confidentiality guarantees privacy so that only the intended data sink can read the data in transit. These safeguarding methods are used for both data in transit and at rest. Access control describes protection schemes that resist unauthorized access to resources. Authentication, Authorization, and Accounting are known as AAA security services, providing the basic framework to control access. Whereas authentication service is the method of verifying the identity of an entity to prevent unauthorized access, authorization service determines which resources entities are entitled to access and their operations. Authorization is accomplished by using an access control list, which determines whether a user has certain access privileges once the user authenticates. In other words, authorization controls what and when an object or entity accesses a specific resource. Accounting refers to logs of connected objects, including access data, length of time of access, and modifications. For instance, a smart grid meter might keep track of the amount of power usage by each connected entity at home over time. The challenge of cybersecurity accounting services is that it tracks and monitors in real-time, and provides auditing results. In Fog-to-things computing, attack detection scheme needs to be real-time as part of an accounting system. Breaches of confidentiality occur when one or more of the safeguarding mechanisms are not provided accordingly. These disclosures might happen when credentials are stolen or cryptography is broken.

The other important security goal in fog ecosystem is integrity. It refers to the goal of cyber security which provides the accuracy, consistency, and trustworthiness of data. Integrity controls ensure that information can be of high quality. Integrity ensures that data should remain unaltered during capture, storage, retrieval, update, and transfer by unauthorized entities. Integrity protection controls used to ensure data integrity could be divided into preventive and detective methods. They include hashing, data validation checks, data consistency checks, and access controls. Protecting data integrity is a major challenge as it depends on how an organization uses data. It ranges from social media and blog posts (low level) to e-commerce data (high level) to healthcare and emergency (critical level).

The third equally important security principles is the availability of data. Availability ensures that systems and networks are responsive, accessible, and meet expected standards. Cyber attacks such as denial-of-service (DoS) [14]–[16] attacks and device failures can prevent access to IoT devices, which threatens the availability of services for a legitimate purpose. Techniques used to enhance availability include redundancy, backups, increased system resiliency, equipment maintenance, and up-to-date OS and software. From cybersecurity point of view, prevention mechanism such as crypto solutions, and monitoring unusual and suspicious network and systems events as a mechanism of detection are crucial.

D. FOG NODES AS A PROTECTIVE SHIELD FOR IoTs

Distributed fog nodes are ideal architectural spots for implementing and deploying security mechanisms. The deployment of security mechanisms at fog nodes could be a protective shield for IoT as this is a complete shift in premise to traditional IT perimeter security [11]. Firstly, the storage of security credentials is better protected than if it is stored in the smart devices, and be more available and up-to-date than if it is maintained in the cloud. This architecture provides the distribution of security services for scalability, and a mechanism for protecting resource constrained IoT devices against sophisticated cyber-attacks. It is also apparent that FC enables to identify attacks and suspicious behaviors quickly as fog nodes are closer to the IoT devices than the cloud. This a mechanism of providing real-time cybersecurity breach incident response services for smart applications, particularly for smart grids, critical industry functions, and smart cities. Suppose that cybersecurity system for smart grid and the connected car is provided by the cloud provider and infected by malware which can completely block the power generators in the grid or engines of the car. In both cases, the complete shutdown of the systems is catastrophic to delay. Furthermore, FC creates an environment of collaborative attack fighting in which IoT devices share attack signatures and experiences For instance, artificial intelligence based lightweight intrusion detection can be implemented on collaborative Fog nodes in collaboration to detect suspicious traffic. Moreover, as data communications are confined to distributed edge networks, FC design protects eavesdropping easier than core networks such as the cloud. Thus, the deployment of security services and functions at fog nodes for IoT applications is an ideal solution.

E. SECURITY MECHANISMS

Cryptographic solutions have been playing major roles in secreting Internet. The most widely used functions of cryptography are authentication and encryption. As it is connected to Internet, Fog-to-things computing inherits threats from traditional Internet. The environment is also prone to zero-day attacks as a result of newly introduced flaws in emerging protocols, workflows and devices. Thus, cybersecurity mechanisms such as authentication, encryption and access control [16] need to be implemented in fog-to-things computing using cryptographic elements.

Mutual authentication has been a fundamental cybersecurity mechanism in securing traditional networks against both internal and external attacks. This control is even more important in fog-to-things computing as trust is utmost needed in this ecosystem because of their large-scale communication. Reference [18] describes authentication as the core layer of security framework in IoT/Fog computing in managing strong identity, non-repudiation and building trust in the ecosystem. By building trust, it is viable to combat threats such as man-in-the-middle, impersonation, and replay attacks, which are the most threatening attacks in IoT/fog computing. Architectural wise, authentication architectures such as 802.1AR/ IEEE 802.1X could be extended for fog systems. However, authentication itself cannot guarantee the delivery of data without eavesdropping or modification by adversaries.

The transportation of data over an insecure channel such as wireless should be guarded by encryption mechanisms. The state of the art of encryption schemes is dependent on cryptographic suites such as Advanced Encryption Suite (AES) for confidential data transport, and Rivest-Shamir-Adleman (RSA) for digital signatures. While the algorithms are robust in fulfilling security requirements, they are not directly suitable for resource-constrained IoT/Fog networks as they require high resource usage. As resources are owned by multi-parties and users, access control also plays a pivotal role in the security of fog computing. Privacy has been always a major concern of Internet, but more crucial for IoT because these devices host applications that trace the location and behaviors of individuals. The good scenarios of privacy issues are health care systems in which medical equipment are tracked, and vital patient data are monitored. In this case, the identity of the device should be known without revealing the identity of the owner. As a combating mechanism, cryptographic elements still play the major role in cybersecurity of Internet and continues to be on the internet of things as well for their robustness. Because of resource limitations of IoT/Fog networks and evolving nature of cybersecurity, however, the traditional cryptographic mechanisms such as RSA fail to support fog-to-things computing. Thus, investigating lightweight cryptographic suites is of great importance.

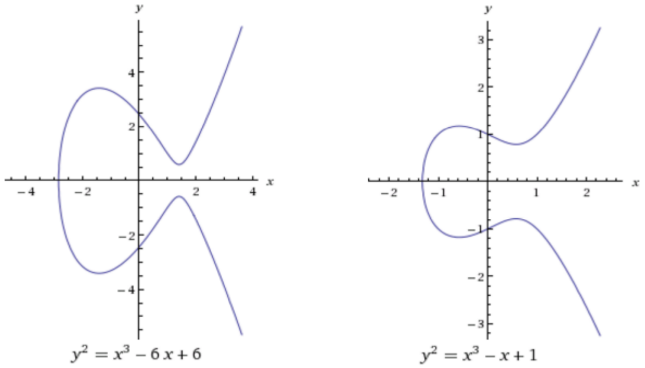

FIGURE 3. Typical elliptic curves.

III. ECC AS A LIGHTWEIGHT CRYPTOGRAPHIC SOLUTION

A. OVERVIEW

Proposed by Koblitz and Miller in the 1980s using group points on an elliptic curve defined over a finite field in discrete logarithmic cryptosystems, Elliptic curve cryptography (ECC) is an algebraic method that uses the properties of elliptic curves to produce cryptographic algorithms. This curve is expressed over a non-singular cubic polynomial equation with two unknowns over a field F in the form of:

$$y^2 = x^3 + ax + b \text{ (mod F), where } 4a^3 + 27b^2 \text{ (mod p)} \neq 0.$$

Elliptic curves have the property that if a straight line that intersects the curve in two points is drawn, it will also intersect the curve in a third point that is either on the curve or the point of infinity. The mirror of this third point over x-axis is the addition of the two points, which is crucial for key generation. The other important property of elliptic curves is that if you have a point P (x, y) then -P will be (x, −y), i.e. two vertical lines that never cross the curve at the third point (P − P = 0) or cross at infinity. In another word, they are symmetric over the x-axis. It has also the domain D = (q, F, a, b, P, n), where q is a prime number, F is the field, a and b are the curve coefficients, P is the base point and n is the order of P. Fig. 3 shows examples of elliptic curves.

The common way to performing multiplication of point in elliptic curves is through point doubling (P + P = 2P). In general, ECC point additions of P (x1, y1) and Q (x2, y2)

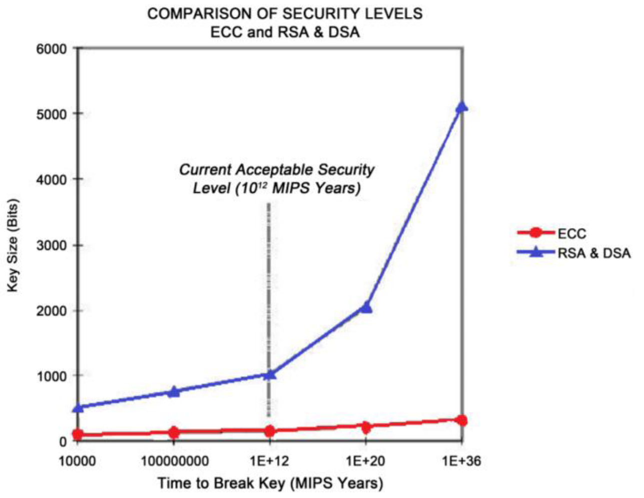

FIGURE 4. Comparison of security levels of ECC and RSA [19].

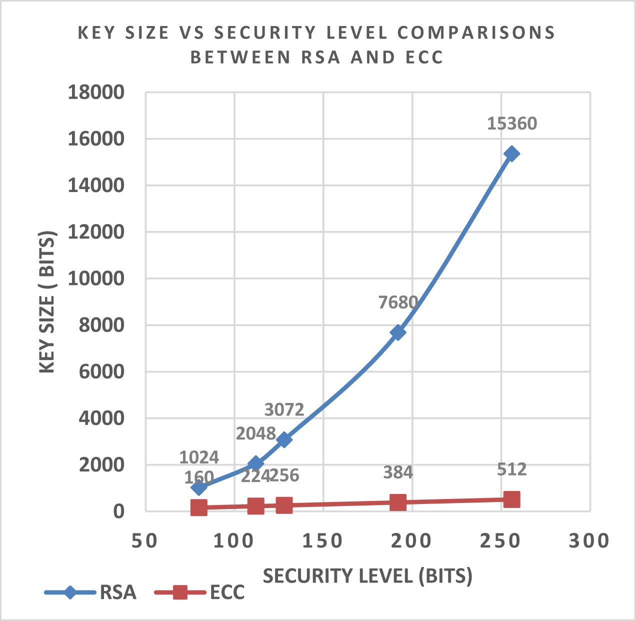

FIGURE 5. Comparison of Key sizes and Security levels between ECC and RSA.

TABLE 1. Comparison of key size, in bits, for RSA and ECC.

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

to yield R (x3, y3) are calculated as $x3 = m2 - x1 - x2$ mod p, $y3 = -y1 + s(x1 - x3)$ mod p, and the slope $m = (y1 - y2)/(x1 - x2)$ mod p [20]–[22].

B. EL GAMAL BASED ECC

In the group G consisting of x and y, it is required that the classical Discrete Logarithm Problem (DLP) solves k from the $x^k = y$. The ECC approach having points P and Q on its curve in group G requires solving k from P o k = Q. The security assumption of ECC is related to Elliptic Curve Discrete Logarithm Problem (ECDLP), where solving the discrete logarithm of a random EC point with respect to a publicly known generator is computationally infeasible. ECC can be used for El Gamal based encryption, ECC Diffie-Hellman based secure key exchange, and authentication and digital signatures.

The original concept of Elgamal lies in embedding a message m in α^k and β^k where α is a root of a large prime p, ka random number, and $\beta = \alpha^a$. The parameters (α, β, p) are public while the sender possesses k and the receiver owns an as a secret key. The message owner sends the pair (α^k, β^km) to the receiver, which decrypts as $(\alpha^k)^{-a*}$ (β^km) = $(\alpha^a)^{-k}$ * (β^km) = $(\beta^{-k})^*$ (β^km) = m. It is difficult for the adversary to solve DLP to get the message or the secret number.

The ECC based Elgamal encryption is the modification of basic Elgamal in which α and β are points on the elliptic curve with multiplications replaced by addition, and multiplication used instead of exponents. An elliptic curve C, a point on curve α (x1, y1), secret integer a of the receiver and random number k of the sender are selected. The point β (x2, y2) is calculated as $\beta = a^*\alpha$. The sender calculates two other points as c(x3, y3) and d(x4, y4) where c(x3, y3) = (x3, k*α) and d(x4, y4) = (x4, m + k *β), and sends to the receiver as (c, d). As the message is carried by the y-coordinates, it also suffices to send as (y3 = k*α, y4 = m + k*β). The decryption proceeds a negative addition process as y4 − (a*y3) = (m + k*β)-a*(k*α) = m + a*k*α − a*k* α = m. The sniffers can intercept the message, but ECDLP makes it infeasible to get k or m [23].

Though RSA and other cryptosystems have been tremendously used in security applications, the current trend of massive growth of IoT applications could necessitate looking at alternative cryptographic solutions that satisfy the nature of these smart devices. RSA assumes that the longer the keys the better it resists against attacks. However, this principle does not hold for small devices which are constrained in processor, memory, and bandwidth. With the prevalence and increment in the number of smart objects, Elliptic Curve Cryptography (ECC) will play a significant role in cryptography as RSA is likely to be unusable with resource-constrained devices. ECC is advantageous in that it provides the same security level with RSA with smaller key sizes but it provides more efficient implementation than RSA [23]. For instance, the security key of 256-bit ECC is assumed to be equivalent to 3072-bit RSA. Table 1 shows that the efficiency of ECC as the increment ratios show that RSA key size should be increased more than double when that of ECC is doubled. In larger key sizes, ECC scheme provides substantial benefits in terms of providing faster encryption/decryption, smaller storage, faster computations and fewer power utilization. Embedded algorithms such as ECC can be adopted for cybersecurity schemes of fog-to-things computing as it lightweight in storage and computations.

C. ECC BASED PROXY RE-ENCRYPTION

The implementation of ECC reduces processing and storage requirements of IoT as the key and encrypted message sizes are much less than cryptographic suites such as RSA. In addition to these novel features employed by ECC, it is also possible to further enhance the processing and storage efficiency of IoT devices by offloading functions of heavy cryptographic elements to fog nodes in the vicinity. As fog nodes act as proxy node between IoT and the cloud or another IoT in fog computing, ECC based proxy re-encryption technique [27], [30] could be adapted for resource-constrained Fog-to-things communication using Fog node as a broker. It is an encryption scheme by which a broker node such as Fog node is provided with intermediate key rk1,2 that enables it to convert original message m encrypted with public key pk1 of a client into an encryption of the same message m under a different public key pk2 without revealing the message contents to the fog node and private keys to either of the parties. Proxy re-encryption protocols have some properties useful for IoTs, namely:

- **Unidirectional**: the re-encryption from A to B does not necessarily imply the reverse.
- **Proxy Transparency** : the existence of proxy should be hidden from clients
- **Key optimal**: the client keys should be kept constant regardless of communicating parties
- **Collusion resistance**: the proxy shouldn't be colluded with any of the clients to retrieve the message or keys of another client
- **Non-transitivity**: if A delegates proxy to re-encrypt for B, the proxy cannot delegate another proxy to re-encrypt for C.

Proxy re-encryption is a new encryption scheme devised for security in a distributed environment such as smart applications supported by the Internet of things. The technique solves the problem of key management and storage limitations of resource-constrained ends such as IoT devices. Instead of using El-Gamal based discrete logarithm problem, we used El Gamal based ECC because of its efficiency in computations. ECC addition of points on an elliptic curve and multiplication of a point on an elliptic curve by an integer shares equivalency to the modulus multiplication and exponentiation in RSA, respectively.

IV. RELATED WORKS

The research on cybersecurity schemes of IoT/Fog computing is in its infancy. Most of the research work on IoT/Fog computing is focusing on architectural issues and application domains, while security schemes have been left to be patched in the years to come. In this section, we thoroughly review and analyze literature and related works in cybersecurity schemes. We review proxy re-encryption related studies for related applications as similar security schemes, to our knowledge, don't exist for fog-to-things applications.

The theoretical and practical aspects of proxy re-encryption have been explored in [24] for distributed file systems. The authors have shown the possibility of using proxy nodes for encryption without disclosing user's data. Yuriy *et al.* [25] have proposed IND-CPA-secure unidirectional Proxy Re-Encryption (PRE) scheme for publish-subscribe applications. The study has demonstrated the efficiency of proxy re-encryption for limited resource embedded systems such as IoT devices. The study in [26] proposed the anonymous key proxy re-encryption which is CPA-secure using the assumption of Decisional Bilinear Diffie Hellman (DBDH). However, the implementation of the scheme is inefficient for resource limited devices. Identity based proxy re-encryption has been explored in [27] to transform plaintexts encrypted under one identity to another. The work has demonstrated the possibility of adapting identity based encryption to proxy re-encryption. Wang [30] proposed id-based proxy re-encryption to protect key leakage from side-channel attacks using fog computing. They have shown that the scheme could be implemented without using PKI certificates. The article [31] summarizes the use of proxy re-encryption for securely sharing data in fog environment. However, the paper lacks detailed experimental evaluations and results.

Our scheme differs from the above works in that we have applied ECC based proxy re-encryption scheme in distributed fog-to-things environment.

V. THE PROPOSED CYBERSECURITY SYSTEM

Though layered cybersecurity mechanisms are required to secure fog-to-things computing, this research concentrates on the encryption mechanism using proxy re-encryption scheme.

A. ALGORITHMS

Our scheme consists of 5 procedures: Key Generation, client encryption procedure, Fog encryption procedure, Fog decryption procedure and client decryption procedure. Since Fog computing is distributed, one specific fog node can be chosen as a trusted authority or coordinator for key and parameters generations. The key generation procedure (procedure 1) produces public curve parameters, public and secret keys of the coordinator fog node. The public elliptic curve parameters PK are sent to the IoT nodes and other fog nodes. The trusted authority securely also sends k_{Ci1} to the IoT device as a private key and (C_i, k_{Ci2}) to the slave fog nodes, where C_i is the identity of client IoT devices. In addition, the trusted key authority has a responsibility of securely storing secret key SK.

Client encryption step shows the encryption of a message m by IoT end using its private key k_{IDi1} as shown in procedure 2. The corresponding fog node re-encrypts the client's ciphertext using the portion of the key on the node k_{Ci2} (procedure 3). Then, the fog node converts the message encrypted by the client to intermediate (intermediate decryption) form so that the cipher can be decrypted only by client C_i as shown in procedure 4. Finally, client C_i decrypts the message using its private key k_{Ci1} as shown in procedure 5.

Procedure 1 Key Generation

```
Input: security parameter $1^n$
Output: public elliptic curve parameters PK and a secret key of master fog node SK.
    1. Two prime numbers p and q are generated in such a way that $q = (p-1)/2$ and $|q| = n$.
    2. A base point P is generated such that cyclic group G is the unique order q of subgroup of $Z_q^*$
    3. Choose k uniformly at random from $Z_q^*$ and compute $h = kP$
    4. Store public curve parameters $PK = (G, P, q, h)$ and $SK = k$
    5. Choose a random $k_{Ci1}$ from $Z_p$ and compute $k_{Ci2} = k \oplus k_{Ci1}$
    6. return $k_{Ci1}$ and $(C_i, k_{Ci2})$
```

Procedure 2 Client Encryption

```
Input: Message m, the public elliptic curve parameters PK, and the client private key $k_{Ci1}$
Output: ciphertext using encryption $E_{IDi}(n)$
    1. Choose r randomly from $Z_q$
    2. $E_{Ci}(m) \leftarrow (rP, m \oplus rk_{i1}P)$
    3. return $E_{Ci}(m)$
```

Procedure 3 Fog Re-Encryption

```
Input: ciphertext $E_{Ci}(m)$, the public elliptic curve parameters PK, and the fog node key $(C_i, k_{Ci2})$ for user $C_i$
Output: The ciphertext E(m)
    1. Compute $(rP)k_{Ci2} \oplus (m \oplus rk_{Ci1}P) = rP(k_{Ci1} \oplus k_{Ci2}) \oplus m = rPk \oplus m$
    2. $E(m) = (rP, rPk \oplus m)$
    3. return E(m)
```

B. SECURITY ANALYSIS

The fog nodes are assumed to be semi-trusted entities in the process of re-encryption while the coordinator node or the trusted authority is trusted in the system by all entities. The communication between things and the fog nodes require several security requirements such as correctness, confidentiality and key scalability. The encrypted message sent from one device to the other should be correctly decrypted. The messages sent by each IoT device should not be disclosed to the fog nodes and unintended IoT devices. The scheme should also eliminate the need to share security keys by decoupling the sender and the receiver since it is not scalable in massively connected IoT environment. IoT devices lack the capability of processing, storing, and communicating security keys with all other clients. In this study, we mainly focus on confidentiality requirement.

Procedure 4 Fog Decryption

```
Input: Encrypted message $E(m) = (rP, rkP \oplus m)$ and the Fog key set $(C_i, k_{Ci2})$ corresponding to client $C_i$.
Output: The intermediate cipher $d_i(m)$ that only can be decrypted by client $C_i$.
    5. Compute $(m \oplus rkP) \oplus (rP)(k_{Cj2}) = rP(k \oplus k_{Cj2}) \oplus m = rP\,k_{Cj1} \oplus m$
    6. $d_i(m) = (rP, rP\,k_{Cj1} \oplus m)$
    7. return di(m)
```

Procedure 5 Client Decryption

```
Input: intermediate cipher from Fog node $d_i(m) = (rP, rP\,k_{Cj1} \oplus m)$ and the client private key $k_{Cj1}$.
Output: message m
    1. $m = (rP\,k_{Cj1} \oplus m) \oplus (rP)(k_{Cj1})$
    2. return m
```

A cryptographic construction is *correct* if the decryption function always produces expected results with proper key. Our scheme has to deal with the correctness of encryption and decryption. Assuming that $c_i \leftarrow Re\_Enc(\ldots, Enc\,(m, \ldots))$ is a ciphertext, $\forall m \in M$, $ID_i, ID_j \in \{0, 1\}^*$, where $\boldsymbol{k_{Ci1}}, \boldsymbol{k_{Ci2}}$ are generated by initialization and key generation procedure, the following holds for correctness of our scheme:

- $Decrypt(k_{Ci1}) = m$
- $Decrypt\ (k_{Cj2}, Re\_Enc(k_{Ci2})) = m$

This indicates that our construction has security correctness as it has been proved by implementation.

A system is said to be **secure** if the probability of breach by adversaries is negligible. The adversary is assumed to be computationally bound random algorithm, and runs in probabilistic polynomial time (PPT) to show that the probability of breach is negligible. A function f is said to be negligible function if for each polynomial p() there K such that for all integers k > K it holds that $f(k) < \frac{1}{f(k)}$. The existence of negligible function is required by a pseudorandom function whose output is indistinguishable from real random function by the adversary. A function $f{:}\{0, 1\}^* \times \{0, 1\}^* \to \{0, 1\}^*$ is pseudorandom if for all probabilistic polynomial time (PPT) adversaries A, there exists a negligible function neg such that $\left|P\left[A^{f_k(.)} = 1\right] - P\left[A^{F(.)} = 1\right]\right| < neg(n)$ where $k \to \{0, 1\}^n$ and F are chosen at uniform random from their corresponding set. The proof lies on the assumption that Diffie-Hellman (DH) algorithm is secure against adversary in the group G, and it is difficult for the adversary to obtain the group components such as kP. The DH algorithm with respect to group G of acyclic order q (|q| = k is secure against a PPT adversary A if there exists negligible function neg such that $|P[A(G, q, P, kP) = 1] - P[A(G, q, P, aP) = 1]| < neg(k)$ for randomly chosen $k, a \in Z_q$.

Our ECC based proxy scheme should be proved to be secure under chosen plaintext attack (IND-CPA) even if proxy encryption has been proved to be indistinguishable under IND-CPA. The cryptographic algorithm is IND-CPA secure if PPT adversary cannot identify the source of an encrypted message which has been taken randomly from two plaintexts with non-negligible probability.

Given that the DH problem in the group G is hard to break, then our ECC based proxy re-encryption scheme FE is IND-CPA secure against the fog node. It means that for a PPT adversary A there exists a negligible function neg such that

$$succ^{A}_{FE,fog}(k) = P\left[\, b' = b \;\middle|\; \begin{array}{c} (PK, SK, k_{\mathrm{Ci1}}, k_{\mathrm{Ci2}}) \leftarrow Init_{gen(1^n, C)} \\ m_0, m_1 \leftarrow A^{FE_{ENC(k_{\mathrm{Ci1}})}}(k_{\mathrm{Ci2}}) \\ b \xleftarrow{R} \{0, 1\} \\ FE_i(m_b) = FE_{ENC(k_{\mathrm{Ci1}}, m_b)} \\ b' \leftarrow A^{FE\_ENC(k_{\mathrm{Ci1}},)}(k_{\mathrm{Ci2}}, FE_i(m_b)) \end{array} \;\middle|\; \right]$$

where C is IoT clients, k_{Ci1} is client key, k_{Ci2} is the re-encryption key.

Proof: Let us assume that PPT adversary A′ tries to solve the ECDH problem used in our ECC based proxy re-encryption using function A. The adversary uses inputs such as G,P,q,h some random r, k. The adversary A′ functions as follows:

- It sends public parameters (G, q, P, h to A)
- Then, by randomly choosing k_{Ci1} from $\mathrm{Z_p}$ for each client IoT, it computes $\mathrm{k_{i2}P = (k \oplus k_{Ci2)}P}$. It stores all $(_{\mathrm{Ci}}, k_{\mathrm{Ci1}}, k_{\mathrm{Ci2}}\mathrm{P})$.
- A tries to pass m to A′, and A′ randomly chooses r from $\mathrm{Z_p}$ and replies with $\mathrm{(rP, m \oplus rk_{i1}P)}$
- A produces $\mathrm{m_0, m_1}$. A′ selects a random bit b and sends $\mathrm{rP, rP\ k_{Ci2} \oplus\ (m_b \oplus rk_{i1}P)}$ to A
- A produces b ′, and If b $= b'$, A′ outputs 1, otherwise 0.

As rP is a randomly created from random r, the adversary gets no information about the value of $\mathrm{m_b}$ from the random element of G setting $\mathrm{rP\ k_{Ci2} \oplus\ (m_b \oplus rk_{i1}P}$. Without any additional information, the adversary A must distinguish between $\mathrm{m_0}$ and $\mathrm{m_1}$. The success probability of b ′ = b is exactly 1/2 when b is chosen uniformly randomly, and A′ outputs 1 iff A outputs b′ = 0, in which case Pr[A′ (G, q, P, rP) = 1] = 1/2. This indicates that our scheme is IND-CPA secure against the adversary and the fog node.

C. PERFORMANCE ANALYSIS AND DISCUSSIONS

It has been discovered that metrics such as runtime, throughput, and ciphertext expansion can be used to evaluate the performance of security functions. Runtime measures the time taken on CPU computations while throughput refers to the bits of plaintext that can be processed per unit time to compute encryption, re-encryption and decryption functions by various nodes in the fog-to-things computing over various security parameters. Moreover, ciphertext expansion represents the number of plaintext bits equivalency in the ciphertext whereas memory usage stands for the size of memory consumption to implement security function of various security configurations.

TABLE 2. Curve parameters.

	Security levels		
Parameters	80 bits	128 bits	256 bits
Length of q	512	1536	7680
Length of r	160	256	512

TABLE 3. Our scheme vs RSA encryption and decryption runtime for 32 bytes of message.

security	Enc (mill.sec)		Dec (mill.sec)		ReInc(mill.sec)
Bits	ECC	RSA	ECC	RSA	ECC
80	3.59	20.7	1.24	198	10.89
128	28.26	30.5	6.48	796	79.34
256	541.79	869	222.33	3876	4077.35

TABLE 4. Our scheme vs RSA encryption and Decryption runtime for 64 bytes of message

Security	Enc (mill.sec)		Dec (mill.sec)		ReInc(millsec)
Bits	ECC	RSA	ECC	RSA	ECC
80	3.59	116.6	1.26	266.6	10.32
128	34.98	143.5	6.82	1086	69.38
256	528.4	1124	239.4	5436	4313.8

TABLE 5. Our scheme vs RSA encryption and Decryption runtime for 128 bytes of message

Security	Enc (mill.sec)		Dec (mill.sec)		ReInc(millsec)
Bits	ECC	RSA	ECC	RSA	ECC
80	4.97	149	1.65	300.6	11.64
128	47.6	205.7	7.568	1286	72
256	503	2870	229.5	6654	4170

Our security scheme has been implemented using Java on the top of nics-crypto [24] to support proxy re-encryption of various parameters. The experimentation has been conducted on a laptop of Intel(R) Core (TM) i7-6700HQ CPU @2.60 GHz with a RAM of 32GB running Windows 10. The experiment ran 20 times for the parameters (sec bits) over certain data sizes to measure runtimes and throughputs of security functions. We measured the execution time and throughput on three categories of message sizes (32, 64 and 128 bytes) for an elliptic curve using three different security levels (80 bits, 128 bits, and 256 bits). The parameters of EC are shown in table 2.

As shown on fig.6, the execution time of proxy-based ECC encryption for fog-to-things communication has been tested on multiple message sizes and security levels.

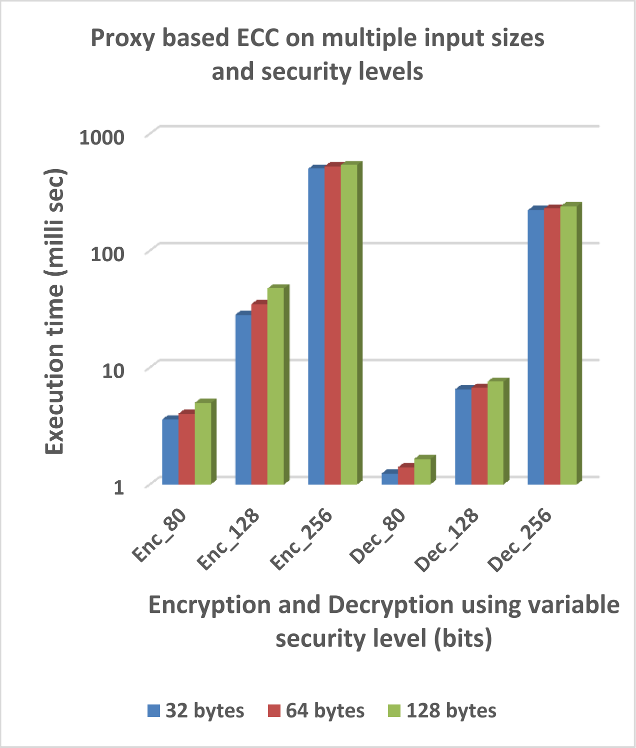

FIGURE 6. Encryption and decryption time of our scheme on multiple data sizes and security levels.

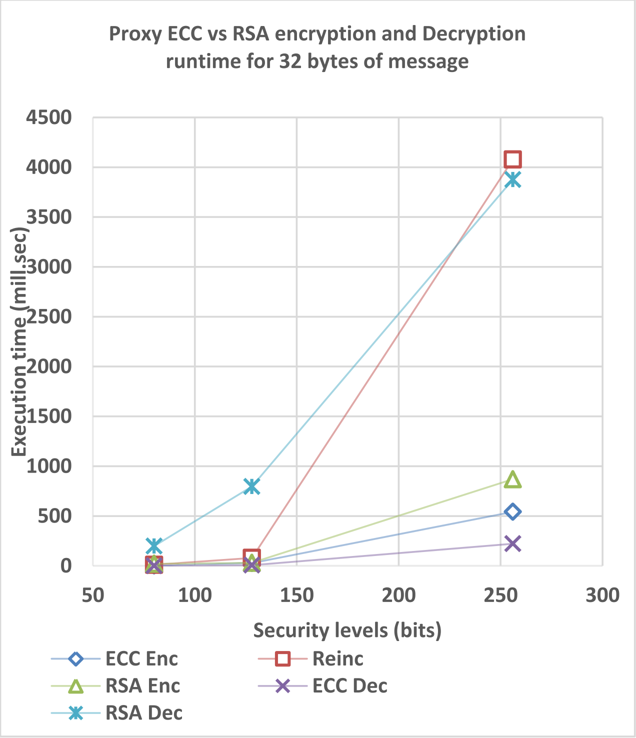

FIGURE 7. Our scheme vs RSA encryption and Decryption runtime for 32 bytes of message.

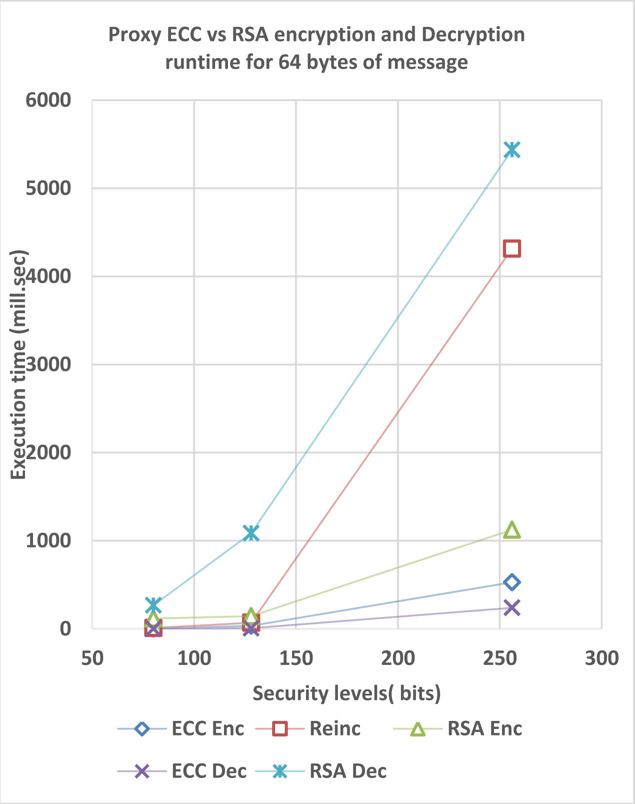

FIGURE 8. Our scheme vs RSA encryption and Decryption runtime for 64 bytes of message.

Though encryption and decryption times increase with increasing security levels, the experiment indicates it almost remains the same for multiple data sizes. However, the time spent in encrypting a given message size using a specific security level is more than double of the time spent for a corresponding decryption.

As it can be observed from fig.7-9, the encryption and decryption of multiple message sizes using proxy ECC are faster than its RSA mechanism. For instance, as shown in the table 3-5, the encryption times of 80, 128,256 bits of security on each data size are each less than that of the corresponding RSA It has been demonstrated that the RSA decryption tends to be the slowest process while ECC decryption has been shown to be the fastest in computations. This indicates that offloading security functions to fog nodes can decrease processing time and resources from resource-limited devices such as IoT. As shown in fig. 10, the encryption throughput of proxy ECC is higher than that of the corresponding RSA security bits over various data sizes as it takes longer for RSA to process similar size messages. Moreover, while the throughputs of our scheme grows with message sizes in a given security level, the throughput of RSA decreases with increasing data sizes per a security level. This emanates from the fast execution of our scheme in encryption and decryption processes. As the experiment conducted on 256 bytes of data shows in table 6, the memory requirements of our approach for the three major cryptographic functions are lower than the corresponding RSA functions in a given security levels. This inculcates the suitability of our approach for memory limited devices.

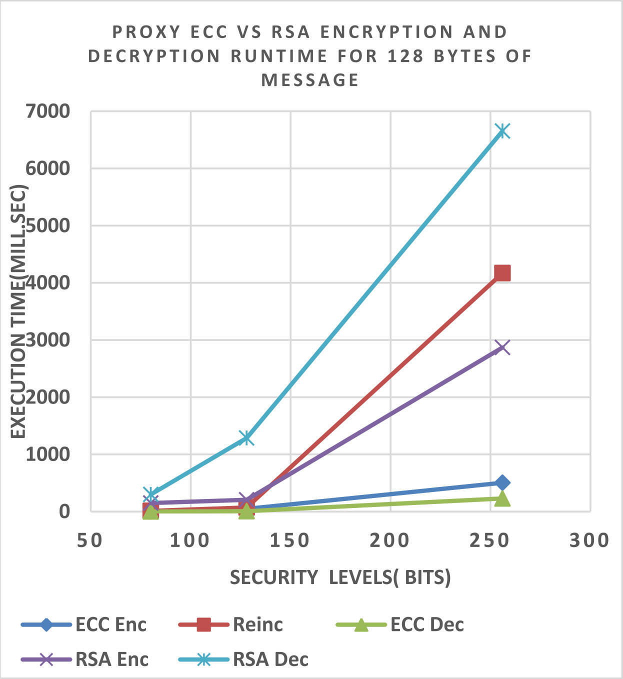

FIGURE 9. Our scheme vs RSA encryption and Decryption runtime for 128 bytes of message.

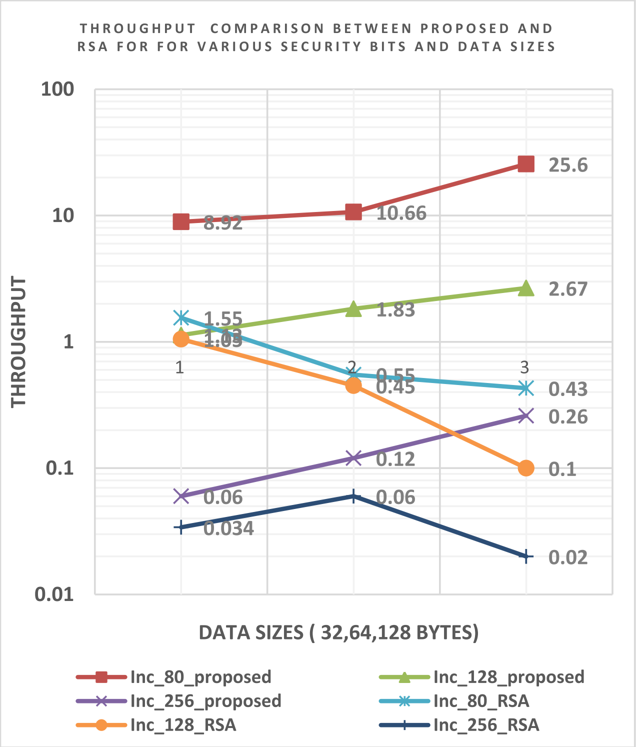

FIGURE 10. Our scheme vs RSA throughput comparison for for various data sizes and security bits.

Apart from the runtime performance, our scheme offers several advantages. From the practicality point of view, the generation of keys and public parameters by the trusted coordinator node makes the smooth accomplishment of encryption and re-encryption processes. The approach is *unidirectional* in that it enables the sender IoT device to delegate a fog node to re-encrypt the ciphertext for the receiver to decrypt, but the reverse is not necessary. The system is also *non-interactive* in that the sender and the receiver don't need to communicate for the construction of re-encryption key. Additionally, plaintext-cipher ratios vary as variable size messages produce constant size (386 bytes) of ciphertext. This is non-expansion makes the approach efficient for embedded devices. The proxy node can produce one-time parameters in scalable manner, and needs to be online for the real-time applications of IoT. On the hand, locality based coordinator nodes could be designed for large networks to make the system more scalable. However, our system is limited to sing-hop fog nodes, and the case of multi-hop nodes is open for future studies. The main time-consuming operations are global parameters generation, which is computed once during registration.

TABLE 6. Our scheme vs RSA memory consumption on 256 bytes of message.

Memory consumption	*Our scheme security level*			*RSA security level*		
	80 bits	**128 bits**	**256 bits**	***80 bits***	***128 bits***	***256 bits***
Encryption	0.01MB	2.16MB	93MB	*0.9MB*	*15.80MB*	*372MB*
Re-encryption	4MB	12.11MB	95MB	*24MB*	*66.7MB*	*380MB*
Decryption	0.015MB	2.5MB	35MB	*1.05MB*	*14.47MB*	*140MB*

VI. CONCLUSION AND FUTURE WORK

This research has proposed ECC based proxy re-encryption for fog-to-things as a lightweight encryption scheme. The security scheme has been analyzed for encryption and decryption runtime efficiency, and throughput ciphertext expansion. The implementation has proved the effectiveness and efficiency of outsourcing security functions to fog nodes for IoT applications. In addition, the implementation of encryption using ECC produced smaller size cipher texts than RSA, proving that ECC is an appropriate cryptographic mechanism for embedded systems such as IoTs. It has been concluded that lightweight security mechanisms for IoTs can be achieved by offloading security functions of IoTs to fog nodes for resource constraints as well as employing ECC for its smaller message sizes. In the future, implementing on real-platforms such as raspery and Arduino will be considered for practical applicability.

ACKNOWLEDGMENT

This work was supported by the Soonchunhyang University Research Fund and also supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2014-1-00720) supervised by the IITP (Institute for information & communications Technology Promotion).

REFERENCES

[1] A. Diro, N. Chilamkurti, and N. Kumar, ''Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog computing,'' *Mobile Netw. Appl.*, vol. 22, no. 5, pp. 848–858, 2017.

[2] (2016). *Securing the Internet of Things: A Proposed Framework*. [Online]. Available: http://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html

[3] A. A. Diro, N. Chilamkurti, and P. Veeraraghavan, ''Elliptic curve based cybersecurity schemes for publish-subscribe Internet of Things,'' in *Proc. Int. Conf. Heterogeneous Netw. Quality, Rel., Secur. Robustness*, Jul. 2016, pp. 258–268.

[4] I. Stojmenovic, ''Fog computing: A cloud to the ground support for smart things and machine-to-machine networks,'' in *Proc. IEEE Austral. Telecommun. Netw. Appl. Conf. (ATNAC)*, Nov. 2014, pp. 117–122.

[5] A. A. Diro, H. T. Reda, and N. Chilamkurti, ''Differential flow space allocation scheme in SDN based fog computing for IoT applications,'' *J. Ambient Intell. Humanized Comput.*, pp. 1–11, Jan. 2018, doi: https://doi.org/10.1007/s12652-017-0677-z

[6] I. Stojmenovic and S. Wen, ''The fog computing paradigm: Scenarios and security issues,'' in *Proc. IEEE Federated Conf. Comput. Sci. Inf. Syst.*, Warsaw, Poland, Sep. 2014, pp. 1–8.

[7] M. Ibrahim, ''Octopus: An edge-fog mutual authentication scheme,'' *J. Netw. Secur.*, vol. 18, no. 6, pp. 1089–1101, 2016.

[8] S. Yi, Z. Qin, and Q. Li, ''Security and privacy issues of fog computing: A survey,'' in *Proc. Int. Conf. Wireless Algorithms, Syst. Appl. (WASA)*, 2015, pp. 685–695.

[9] I. Yaqoob *et al.*, ''The rise of ransomware and emerging security challenges and solutions in the Internet of Things,'' *Comput. Netw.*, vol. 129, no. 2, pp. 444–458, Dec. 2017.

[10] A. Diro and N. Chilamkurti, ''Distributed attack detection scheme using deep learning approach for Internet of Things,'' *Future Generat. Comput. Syst.*, vol. 282, pp. 761–768, May 2017. [Online]. Available: https://doi.org/10.1016/j.future.2017.08.043

[11] *Top 5 Ways Fog Computing Can Make IoT More Secure*. Accessed: Nov. 12, 2017. [Online]. Available: https://www.openfogconsortium.org/top-5-ways-fog-computing-can-make-iot-more-secure/

[12] J. McCumber, *Assessing and Managing Security Risk in IT Systems: A Structured Methodology*. 1st ed. Boca Raton, FL, USA: Auerbach Publications, Jun. 2004.

[13] W. Stallings and M. P. Tahiliani, *Cryptography and Network Security: Principles and Practice*, vol. 6. London, U.K.: Pearson, 2014.

[14] *Distributed Denial of Service Attacks—The Internet Protocol Journal*. Accessed: Nov. 14, 2016. [Online]. Available: http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/dos-attacks.html

[15] C. Patrikakis, M. Masikos, and O. Zouraraki, ''Distributed denial of service attacks,'' *Internet Protocol J.*, vol. 7, no. 4, pp. 1–2, 2004.

[16] S. Yi, C. Li, and Q. Li, ''A survey of fog computing: Concepts, applications and issues,'' in *Proc. Workshop Mobile Big Data (Mobidata)*, New York, NY, USA, 2015, pp. 37–42. [Online]. Available: http://dx.doi.org/10.1145/2757384.2757397

[17] C. Dsouza, G. J. Ahn, and M. Taguinod, ''Policy-driven security management for fog computing: Preliminary framework and a case study,'' in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integr. (IEEE IRI)*, Redwood City, CA, USA, Aug. 2014, pp. 16–23, doi: 10.1109/IRI.2014.7051866.

[18] I. Stojmenovic, X. Huang, and H. Luan, ''An overview of fog computing and its security issues,'' *Concurrency Computat. Pract. Exper.*, vol. 28, no. 10, pp. 2991–3005, 2016, doi: 10.1002/cpe.3485.

[19] R. Roman, J. Lopez, and M. Mambo, ''Mobile edge computing, fog: A survey and analysis of security threats and challenges,'' *Future Generat. Comput. Syst.*, vol. 78, no. 2, pp. 680–698, Nov. 2016, [Online]. Available: http://dx.doi.org/10.1016/j.future.2016.11.009

[20] *RSA vs ECC Comparison for Embedded Systems*. Accessed: Nov. 12, 2017. [Online]. Available: http://www.atmel.com/Images/Atmel-8951-CryptoAuth-RSA-ECC-Comparison-Embedded-Systems-WhitePaper.pdf

[21] D. Hankerson, S. Vanstone, and A. J. Menezes, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer-Verlag, 2004.

[22] S. Sandeep, ''Elliptic curve cryptography for constrained devices,'' Ph.D. dissertation, Faculty Elect. Eng. Inf. Technol., Ruhr-Univ. Bochum, Bochum, Germany, 2006.

[23] *Elgamal Encryption Using Elliptic Curve Cryptography*. Accessed: Nov. 11, 2017. [Online]. Available: https://cse.unl.edu/~ssamal/crypto/EEECC.pdf

[24] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, ''Improved proxy re-encryption schemes with applications to secure distributed storage,'' *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, 2006.

[25] P. Yuriy, R. Kurt, S. Gyana, and V. Vinod, ''Fast proxy re-encryption for publish/subscribe systems,'' *ACM Trans. Privacy Secur.*, vol. 20, no. 4, p. 14, 2017.

[26] G. Ateniese, K. Benson, and S. Hohenberger, ''Key-private proxy re-encryption,'' in *Topics in Cryptology—CT-RSA* (Lecture Notes in Computer Science), vol. 5473, M. Fischlin, Eds. Berlin, Germany: Springer, Berlin, 2009.

[27] M. Green and G. Ateniese, ''Identity-based proxy re-encryption,'' in *Applied Cryptography and Network Security*. Berlin, Germany: Springer, 2007.

[28] Z. Ali, M. Imran, M. Alsulaiman, T. Zia, and M. Shoaib, ''A zero-watermarking algorithm for privacy protection in a voice disorder detection system,'' *Future Gener. Comput. Syst.*, vol. 82, no. 5, pp. 290–303, 2018.

[29] I. Yaqoob *et al.*, ''Internet of Things architecture: Recent advances, taxonomy, requirements, and open challenges,'' *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 10–16, Jun. 2017.

[30] Z. Wang, ''Leakage resilient ID-based proxy re-encryption scheme for access control in fog computing,'' *Future Generat. Comput. Syst.*, to be published.

[31] Y.-J. Song and J.-M. Kim, ''Secure data sharing based on proxy re-encryption in fog computing environment,'' in *Proc. Asia–Pacific Appl. Sci. Eng. Better Hum. Life*, 2016, pp. 52–56.

[32] M. Sepehri, and A. Trombetta, ''Secure and efficient data sharing with atribute-based proxy re-encryption scheme,'' in *Proc. 12th Int. Conf. Availability, Rel. Secur.*, 2017, p. 63.

ABEBE ABESHU DIRO received the M.Sc. degree in computer science from Addis Ababa University, Ethiopia, in 2010. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Technology, La Trobe University, Bundoora VIC, Australia. From 2007 to 2013, he was the Director of ICT Development with Wollega University, where he is currently a Lecturer in computer science. His research interests include software defined networking, Internet of Things, cybersecuirity, advanced networking, machine learning, and big data.

NAVEEN CHILAMKURTI received the Ph.D. degree from La Trobe University, Melbourne, VIC, Australia. He is currently a Cybersecurity Program Coordinator with the Department of Computer Science and Information Technology, La Trobe University. His current research areas include intelligent transport systems, smart grid computing, vehicular communications, vehicular cloud, cyber security, wireless multimedia, wireless sensor networks, and mobile security.

YUNYOUNG NAM received the B.S., M.S., and Ph.D. degrees in computer engineering from Ajou University, South Korea, in 2001, 2003, and 2007, respectively. From 2007 to 2010, he was a Senior Researcher with the Center of Excellence in Ubiquitous System. From 2010 to 2011, he was a Research Professor with Ajou University. He also spent time as a Post-Doctoral Researcher at the Center of Excellence for Wireless and Information Technology, Stony Brook University, NY, USA, from 2009 to 2013. From 2013 to 2014, he was a Post-Doctoral Fellow with the Worcester Polytechnic Institute, Worcester, MA, USA. In 2017, he was the Director of the ICT Convergence Rehabilitation Engineering Research Center, Soonchunhyang University, where he is currently an Assistant Professor with the Department of Computer Science and Engineering. His research interests include multimedia database, ubiquitous computing, image processing, pattern recognition, context-awareness, conflict resolution, wearable computing, intelligent video surveillance, cloud computing, biomedical signal processing, rehabilitation, and healthcare system.