IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 21, 2021, accepted July 12, 2021, date of publication July 15, 2021, date of current version July 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097267

LiSP-XK: Extended Light-Weight Signcryption for IoT in Resource-Constrained Environments

TAI-HOON KIM[1], (Member, IEEE), GULSHAN KUMAR[2], (Member, IEEE), RAHUL SAHA[2], (Member, IEEE), WILLIAM J. BUCHANAN[3], TANNISHTHA DEVGUN[4], AND REJI THOMAS[5]

[1]Glocal Campus, Konkuk University, Seoul 27478, South Korea
[2]School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab 144411, India
[3]Blockpass ID Laboratory, Edinburgh Napier University, Edinburgh EH11 4DY, U.K.
[4]Punjab Institute of Technology, Punjab Technical University (Main Campus), Kapurthala, Punjab 144603, India
[5]Division of Research and Development, Lovely Professional University, Phagwara, Punjab 144411, India

Corresponding author: Rahul Saha (rsahaaot@gmail.com)

ABSTRACT There is an increasing drive to provide improved levels of trust within an Internet-of-Things (IoTs) environments, but the devices and sensors used tend to be limited in their capabilities for dealing with traditional cryptography methods. Resource constraints and security are often the two major concerns of IIoT (Industrial IoT applications and big data generation at the present time. The strict security measures are often not significantly resource-managed and therefore, negotiation normally takes place between these. Following this, various light-weight versions of generic security primitives have been developed for IIoT and other resource-constrained sustainability. In this paper, we address the authentication concerns for resource-constrained environments by designing an efficient authentication protocol. Our authentication scheme is based on LiSP (light-weight Signcryption Protocol); however, some further customization has been performed on it to make it more suitable for IIoT-like resource-constrained environments. We use Keccack as the hash function in the process and Elli for light-weight public-key cryptography. We name our authentication scheme: *Extended light-weight Signcryption Protocol with Keccack* (LiSP-XK). The paper outlines a comparative analysis on our new design of authentication against a range of state-of-the-art schemes. We find the suitability of LiSP-XK for IIoT like environments due to its lesser complexity and less energy consumption. Moreover, the signcryption process is also beneficial in enhancing security. Overall the paper shows that LiSP-XK is overall 35% better in efficiency as compared to the other signcryption approaches.

INDEX TERMS Internet, IoT, security, authentication, signcryption, attacks.

I. INTRODUCTION

IoT is one of the most promising technologies to be adopted in multi-dimensional applications [1]–[3]. Automation industries, smart developments, vehicle networks, ubiquitous and pervasive computing paradigm often use IoT as backbone [4]–[6]. Even though IoT has provided many advantages, there are some serious problems that relate to security, privacy, compatibility and complexity [7]. As IoT has extended to Industrial IoTs (IIoTs), smart developments and other progressive dimensions, the number of connections has been significantly increased. This increasing number of devices in IoTs pose severe security challenges. For example, IIoT devices may transmit machine status and where fitness devices deal with users' personal data and health data. Therefore, security methods must be applied at the very first level where the devices connect to the network and generate data. Intrusion detection systems are well configured in various networks to prevent third-party hacking of big data from clouds [8]. At the core of security is: Confidentiality; Integrity; and Availability (CIA). However, the other security services such as authentication, authorization, access control non-repudiation are also important. Authentication is the first step to support the security of an environment. This is because once a proper authentication is done, only legitimate users will have the access to certain data; if authentication is not properly implemented, any unauthorized user can enter the network and could undertake malicious activities. Moreover,

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Khurram Khan.

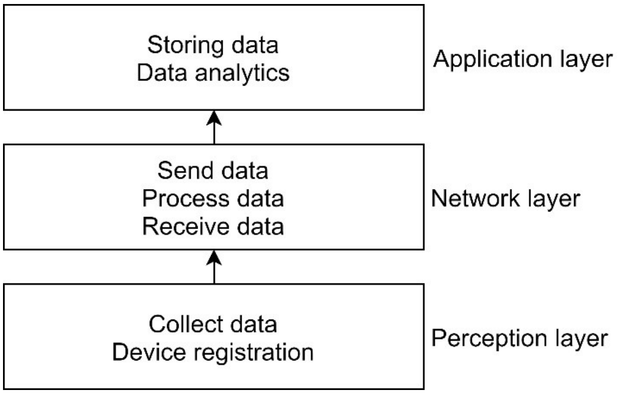

FIGURE 1. Three-Layer architecture of IoT.

the need of short keysize and ciphertexts in authentication is required for resource constrained devices [9], [10].

Therefore, in this paper, we have shown the extended design of an authentication scheme.

A. IoT ARCHITECTURE

To apply a security method such as an authentication scheme, it is often necessary to understand the underlying architecture of the system. The standard architecture of IoT can be configured as three-layer architecture, but, depending on specific requirements, the architecture can have five layers [11], [12]. The three-layer architecture consist of [13] (Figure 1):

- **Perception Layer:** The end-node devices are registered in the network in this layer. Devices such as mobile phones, sensors, RFIDs, and GPS equipment collect data and transfer it to the network layer.
- **Network Layer:** This layer helps in collecting data from the perception layer and sends it to the application layer. This layer provides the connectivity to the whole network. Security controls such as firewalls and intrusion detection systems (IDSs) can be added for security provisions. Communication technologies such as 4G, 5G and beyond, ZigBee, WiFi are used for networking in this layer.
- **Application Layer:** This layer is used to provide the interfacing between an IoT network and the users. It also uses big data analytics for understanding the application behaviour.

Some researchers have also used five-layered architecture. It consists of the basic three layers (as mentioned above) and additionally adding **processing layer** and a **business layer**. The processing layer works an intermediate layer and helps in reserving, observing, transferring and receiving data from other layers. The business layer works as a management layer for the IoT infrastructure, and where the services of actual network are actually being managed. Considering the layers of architectures, we notice that devices connect in the perception layer. If this layer can be protected with strong and effective security approaches, the other layers can be further protected. There are many attacks that are exposed with IoT, including Denial-of-Service (DoS), Man-in-the-Middle attack (MiM), identity-based attacks, and cloning attacks [14]. One way to handle the security issues at the device level is to use efficient encryption and authentication protocols. Encryption provides confidentiality of the data while transferred from the perception layer to any other of the layers, and authentication provides the verifiability of the users' identities. Various algorithms exist with generic cryptographic constructions for both encryption and authentication. However, these methods are not always feasible for IoT environments due to their complexities and poor resource utilization. Moreover, efficiency also reduces and costs can be high.

One enhanced method is *signcryption* [15]. As the name identifies, it digitally signs the message and encrypts it in a *single logical step*, where the digital signature provides the authentication of the user's device at the perception layer. Such signcryption schemes are beneficial on resource-constrained scenarios due to their reduced computational cost and effectiveness as compared to the *sign-then-encrypt* approach. In 1997, Zheng introduced signcryption [15] and uses elliptic curve cryptography (ECC). The basic working of signcryption is shown in Figure 2. With the increasing demand of the efficient cryptographic algorithms, signcryption has also extended to light-weight signcryption algorithms, and where the researchers further reduce the complexity of the algorithms while maintaining the justified security notions [1].

B. CONTRIBUTION

Signcryption methods are already existing. Lightweight signcryption is also emerging. However, the continuous development of the embedded devices and tiny sensors urge the development of ultra-light versions of the protocols. This motivates us for the present research to extend the lightweight property of our previously developed LiSP [13]. The main contributions of this paper are:

1) We address the resource constraints of the IoT applicability by the design of an extended light-weight authentication.
2) We extend our previous work of LiSP [13] and modify the hash with Keccak. This is customized to the modules as per the suitability of the operational environment. Moreover, we use Elli encryption which is itself a light weight. Thus, LiSP is lighter than previous.
3) We perform extensive simulation on our proposed signcryption and the state-of-the-art schemes. The comparative study confirms the superiority of LiSP-XK over others based on complexity and energy consumption.

C. PAPER ORGANIZATION

The rest of the paper has been organized as follows. Section II reviews some recent developments of light-weight protocols for IoT authentication. Section III shows the proposed work. Section IV shows the results and Section V concludes the paper.

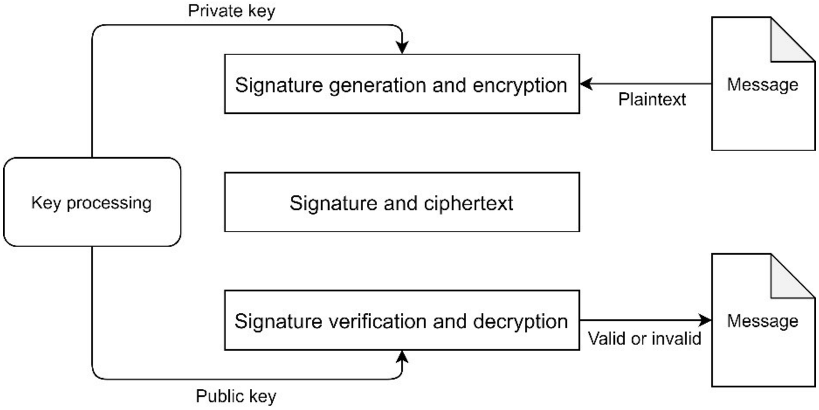

FIGURE 2. 3-Layer architecture of IoT.

II. RELATED WORK

The expanded applications of IoTs has drive demand for new methods of security. Moreover, the resource constraints of ubiquitous and pervasive computing, Wireless Sensor Networks (WSNs) and Industrial IoTs (IIoTs) often require less complex methods with more security performance. There are various authentication schemes available in IoTs; biometrics and two factor authentications are more in use out of all [16]. However, two factor authentications face cryptanalysis problems and need further improvement [17]. In this section, we review some of the state-of-the-art methods of authentication protocols. We segregate the methods in two parts: existing light-weight authentication protocols; and signcryption and light-weight signcryption techniques.

A. LIGHT-WEIGHT AUTHENTICATION PROTOCOLS

The SPRA (Scalable Pseudo-random RFID private mutual Authentication) protocol [18] uses a Pseudo-Random Number Generator (PRNG) and secures systems against various attacks. The Dass-Om method is light-weight RFID authentication protocol which uses PRNG and hash functions [24]. This method had authentication problems and was updated [25]. An extended version of [25] is discussed in [26].

For RFID tag security, Gui and Zhang [19] describe a light-weight authentication method that authenticates the RFID device, the RFID reader and the database. Moosavi *et al.* [20] defines a light-weight authentication that uses ECC and D-Quark hashing techniques. This protocol reduces communication overhead and also provides less memory consumption. A session key attack resistant authentication scheme is shown in [21]. A token-based light-weight authentication protocol - Dynamic Token-based Authentication Protocol (DTAP) [22] uses three component requests, a token and a response. Unfortunately, it does not provide security against confidentiality attacks. In [23], a hash-based light-weight mutual authentication protocol where RFID tags move within clusters, and each cluster authenticates every tag that enters its cluster.

An OTP-based authentication method is researched in [27], and a key-agreement emphasized light-weight authentication protocol in [28]. In [29], the authors use intuitive hashing with XOR technique to make the protocol light-weight. Users also have privilege to update the password. A different type of light-weight authentication protocol has been defined in [30], and where whole cryptographic function is based on IKv2 (Internet Key Exchange version 2) and uses block chaining using Advance Encryption Scheme (AES) in Cipher Block Chaining (AES-CBC) mode, along with ECDH (Elliptic Curve Diffie Hellman) for the key exchange.

The work proposed by authors in [31] is a light-weight authentication protocol based on various techniques such as ECC, PRNG, and public-key cryptography. The main feature of this protocol is that it provides mutual authentication for both server and tags. However, it fails to secure the network from DDoS attacks. In [32], the authors show a light-weight key agreement mutual authentication protocol. The main feature of this protocol is to secure the data from an unauthorised user of the cloud server. With security validation, the authors use Burrows-Abadi-Needham (BAN) logic.

In [33], the authors propose a light-weight mutual authentication protocol called as Attestation and Authentication of Low Resources Things (AAoT). This protocol is based on Physical Unclonable Function (PUF). PUFs help in filling memory randomly. This approach helps in resource constraint usage and also helps in the creation of trust zone for every physical device in the network. This is known as *PUF-based Root-of-Trust* (PUFRoT). On the other hand, AAoT fails to secure the network form collision and impersonation attacks.

The major applications of IoT have been observed in healthcare and medical things, smart grids, and drone networks. Almost all the application fields of IoTs have researched some dimension of light-weight authentication schemes in recent years. For healthcare field, the major contributors of authentication schemes include: a *light-weight authentication protocol for cloud-based health-care systems* (LAPCHS) [34]; a two-factor authentication scheme for Wireless Body Area Network (WBAN) [35]; Secure and light-weight RFID authentication protocol for Medical IoT (SecLAP) [36]; a light-weight mutual authentication and key agreement protocol for remote surgery application in tactile internet environment [37], and authentication for e-healthcare using 5G [38]. Some other health-related approaches are also seen in [39]–[41].

Smart grid-based applications of light-weight authentication are observed in [42]–[45]. Other important research directions on light-weight authentications are well configured in [46]–[53]. Though the developed approaches provide enhanced light-weight methods, there is still scope for improvement to reduce the complexity and time consumption of the methods by integrating the concept of signcryption.

B. LIGHT-WEIGHT SIGNCRYPTION

In our previous work [13], we developed a signcryption protocol using the PHOTON hash [68], and showed that the number of iterations and keysize is less as compared to other generic protocols. In [54], a signcryption-based protocol uses heterogenous environment with online and offline signcryption mode. The offline phase does the major computational tasks and the online phase deals with low computing tasks. A hyper-elliptic curve is also used in signcryption as shown in [55] and uses the Hyper-Elliptic Curve Discrete Logarithm Problem (HECDLP). Another ECC-based light-weight signcryption protocol is discussed in [56]. A recent development for light-weight signcryption for WBAN is observed in [57], while a certificate-based option is outlined in [58].

For vehicular networks and trusted IoT, there has been an increase in the usage of signcryptions, such as in [59] and [60]. Industrial applications are also resource-constrained, and one such development is discussed in [61]. Some other signcryption developments are observed in [62]–[64].

From the above discussion of the existing light-weight protocols and the technology prediction of connected devices, it is required for the development of efficient light-weight signcryption protocols. The reason for choosing signcryption for our present research is due to its simplicity of execution compared to *sign-then-encryption* process. We also consider our previous work [13] for our present extended version. In the paper, we customize LiSP with Keccak hash. The major contribution of the presented work is:

- The use of Keccak as a light-weight sponging method.
- The use of hashchain in Keccack.
- The reduction of number of keys, time consumption and memory consumption.

III. PROPOSED WORK

We assume LiSP-XK to be applicable for client-server applications and peer-to-peer network, and where the key generator is secure and untampered. A distributed key generator system can also be utilized like distributed public-keys; however, in the present work we concentrate on LiSP and the out mode of extension and customization, as per the requirement of better light-weight property as compare to LiSP. Keys generated by the key generator are shared through a secure channel. The sender then signs the message by finding the hash of the message and encrypts it using their private key. Here, we use the hashchain using a random number of iterations. Signing and encryption of message are done in a single step under signcryption process. When the receiver receives the signcrypted text, they first decrypt the message to find its hash. This hash value is compared with the value found by unsigning the message. If both the hash values are matched, the message is authenticated otherwise the transaction is discarded. The proposed signcryption scheme for authentication in IoT consists of three phases: i) setup and key generation phase; ii) signcryption phase; and iii) unsigncryption process. In the following subsections, we discuss each of the phase in detail.

A. SETUP AND KEY GENERATION

We use the Key Generator and Verifier (KGV) module as mentioned in our previous work [13]. The master keys are denoted as master private key (M_{pvr}) and master public-key (M_{pub}). The sender's and receiver's keys are used with master keys, hashchain values and random numbers. For the hash, we use the Keccak sponge construction with the specification of Parallel Keccak-f[1600] [65]. The reason of choosing the Keccak family for the hash is due to its memory utilization capacity. Moreover, it avoids a feed-forward mechanism and saves a lot of memory registers at the cost of an invertible iterative process. It also helps in providing a lower (second)-preimage security for the same internal state size. We use this hash function for a hashchain.

The hashchain is constructed with the help of a random number generated from the SRFG [66]. This random number provides the random number of iterations on a secret seed value given as the input to SRFG. The last value of the hashchain is considered as the *public salt* and used for further process. The public salts changes after few transactions and is decided by KGV. The secret seed value is chosen on a hyper-elliptic curve given by $y^2 + h(x)y = f(x)$, and where $f(x)$ is a polynomial degree $n = 2g + 1 > 4$ or $n = 2g + 2 > 4$ with n distinct roots and $h(x)$ is a polynomial of degree $< \ g + 2$. g is the genus >1. We use ρ_1, ρ_2 as the group generators of an additive and a multiplicative cyclic group G_1 and G_2 from the given curve. Both the sender and the receiver have their identities. We generalize it as a node ID denoted as N_{ID}. This identity is further used to generate the public and private keys. The master private and public-keys are given as (M_{pvr}, M_{pub}). The private key and public-key of nodes are given as $\{N_{ID_{pvr}}\}$, $\{N_{ID_{pub}}\}$. For the simplicity of expression, we use N_s and N_R to represent sender and receiver, respectively. The steps of the key generation process is summarized in Algorithm 1.

Thus, the public-private keypair for sender becomes: $N_{S_{pub}}$, $N_{S_{pvr}}$ and the public-private keypair for receiver becomes:

Algorithm 1 Key Generation

```
Input: $N_{ID}$, $Salt$
Output: $N_{ID_{pvr}}, N_{ID_{pub}}$
Select a random number $\tau$ on $y^2 + h(x)y = f(x)$
$M_{pvr} = H(S_{ID}).\tau$ where, $S_{ID}$ is the identity of the KGV or trusted server
$M_{pub} = H(S_{ID}).\rho_1$
$N_{ID_{pvr}} = (M_{pvr}.Salt)||H(N_{ID})$
$N_{ID_{pub}} = M_{pub}||H(N_{ID})$
```

$N_{R_{pub}}, N_{R_{pub}}$. These keys are used further for signcryption and unsigncryption process.

B. SIGNCRYPTION PHASE

The process of signcryption is summarized in Algorithm 2. With this, the sender initiates the signcrypts with a message before sending it towards receiver. The sender chooses λ randomly from (.) operation of $G_1.G_2$. It then computes: $\mathcal{R}$ using inverse λ function; its private key $N_{S_{pvr}}$; and group generators $\rho_1.\rho_2$. The message (M) is concatenated with λ and we encrypt it with receiver's public-key $N_{R_{pub}}$. For this We use Elli for the encryption process [67]. We then use the Keccack hash for $\{C, R, t\}$ where t is the timestamp of the operation. Finally, the transmitting message becomes as $\mathcal{C} = \{C, R, \tau\}$ and it is sent to the receiver.

Algorithm 2 Signcryption Phase

```
Input: $G_1, G_2, \rho_1, \rho_2, M$
Output: $\mathcal{C}$
Select $\lambda \in G_1.G_2$
$R = \lambda^{-1}(N_{S_{pvr}} + q)$,where $q = \rho_1.\rho_2$
$C = E(\lambda||M||t)^{N_{R_{pub}}}$
$\tau = H(C||R)$
$\varphi = (\lambda.\lambda)\ mod\ q$
return $\mathcal{C} : \{R, C, \tau\}$
```

The full signcrypted message is $\mathcal{C} = \{R, C, \tau, \varphi\}$.

C. UNSIGNCRYPTION PHASE

The process of signcryption is summarized in Algorithm 3. With this, the receiver initiates the unsigncryption process once it receives $\mathcal{C}$. The first step in this phase is matching of hash on the received components. It calculates τ' by using the Keccack hash function on $(C||R)$. If τ' is matched with received τ, $\mathcal{C}$ is processed further, otherwise it is discarded due to an integrity failure. The receiver then decrypts $\mathcal{C}$ using its private key $N_{R_{pvr}}$ and obtains the message M, the value of λ, and the value of t. If this $t < t_{arr} + \delta$ (where t_{arr} is the arrival time and δ is the system synchronization error), unsigncryption process execution continues, else the process is aborted due to the stale timestamp invalidity. The receiver calculates φ' with modulo operation on $(\lambda.\tau')$. If the calculated φ' is equal to the received φ, the message is properly validated and accepted.

Algorithm 3 Unsigncryption

```
1:  Input: $\mathcal{C}$
2:  Output: Acceptance or rejection
3:  $\tau' = H(C||R)$
4:  If $(\tau' == \tau)$
5:  Go to step 6
6:  Else
7:  Discard $\mathcal{C}$
8:  $\{M, \lambda, t\} \leftarrow D(C)^{N_{R_{pvr}}}$
9:  if $(t < t_{arr} + \delta)$
10: Goto Step 13
11: Else
12: Abort process ();
13: $\varphi' = (\lambda.\tau')\ mod\ q$
14: if $(\varphi' == \varphi)$
15: $M$ is validated with signcryption and accepted
16: Else
17: $M$ is discarded
```

IV. DISCUSSION AND ANALYSIS

In this section we describe the experimental setup to study the LiSP-XK performance based on complexity, memory consumption and time consumption. For each of the experimental study, we use 50 rounds of operations and we represent the average value. The modularization of the method is also analysed. We compare LiSP-XK with our previous work LiSP [9] and other three existing approaches as mentioned in Liu *et al.* [57], Zhou *et al.* [58], and Elkhalil *et al.* [59].

A. EXPERIMENTAL CONFIGURATION

For the experiments, we consider 20 nodes (devices) including 10 mobile phones and 10 laptops. We choose these devices due to their resource constraints as the primary goal of the light-weight systems being compatible with these devices. The wireless medium uses Wi-fi 802.11 a/b/g/n/ac. The considered memory size of the devices varies from 4 GB to 10 GB to check the memory utilization. Cisco IC3000 Industrial Compute Gateway is used as the server/workstation whose processor is 4-core Intel Rangeley with 8 GB DRAM and RJ-45 traditional console connector. We also check the LiSP-XK validity in Automated Validation of Internet Security Protocol and Applications (AVISPA) and it has been found that LiSP-XK is safe. The hardware and software specification is listed in Table 1.

B. COMPARATIVE RESULTS

We execute all the existing works and our proposed LiSP-XK on the same platform and experimental setup as mentioned in Table 1. We compare the results based on complexity, energy consumption and memory consumption to check the light-weightness of LiSP-XK. We first show the steps involved for the authentication in the processes. The comparison is shown in Table 2. Note that the steps mentioned in Table 2 are not the same as per the algorithms. We mention

TABLE 1. Experimental specification.

Hardware		
Nodes	Number of nodes	Specification
Workstation (Cisco IC3000 Industrial Compute Gateway)	1	4-core Intel Rangeley 8-GB DRAM (soldered down) RJ-45 traditional console connector Hardware-based anti-counterfeit, anti-tamper chip Mean Time Between Failure (MTBF): - IC3000-2C2F 376,580 hours
Laptops	10	Octa-core 1.6 GHz Cortex-A53 Wi-Fi 802.11 a/b/g/n/ac, dual-band, Wi-Fi Direct, Bluetooth RAM 4 GB, storage considered (4GB-10GB)
Mobile phones	10	Octa-core processor, 3.1 Ghz RAM (4 GB), Memory 64GB Wireless connectivity (802.11 b/g/n and Bluetooth 5.0)
Software		
Elli	For light-weight public-key encryption	
Parallel Keccak-f[1600]	For light-weight hashing	

TABLE 2. Authentication process comparison (single authentication).

		Liu et al. [57]	Zhou et al. [58]	Elkhalil et al. [59]	LiSP [13]	LiSP-XK
Key generation steps		6	7	6	4	4
Authentication steps	Signature creation	8	7	6	7	6
	Verification	6	7	6	6	5
Total processes (steps)		20	21	18	17	15

the steps as per the process perspectives. From this table, it is clear that LiSP-XK is executable in a lower number of steps of the process. This perspective leads to the efficiency of LiSP-XK towards the light-weight direction. The next study that we have executed is to measure the communication cost and computation cost. We compute the communication cost in terms of bits and computation cost in terms of process complexity. The bits measurement are as follows. The unique identities use 160 bits, the random number uses 128 bits, the hash outputs (Keccak) are of 256 bits, and the timestamp is of 64 bits. We use a message size of 1024 bits.

The total cost for a single key exchange between two nodes becomes (160 + 256 + 256) and which is 672 bits. So, the communication cost for key exchange becomes (672 + 672 + 128) and which is 1472 bits. Now, to transfer the signcrypted message C, the communication cost of $\mathcal{C} = 256+160+256+128+64$, and which is 864 bits. As a result of communication costs including key exchange and message exchange for LiSP-XK is (1472 + 864) we get 2,336 bits.

We have used 100 iterations for the measurement of communication cost and then the average value is considered. Table 3 shows the comparison of communication cost for all the signcryption schemes under observation. It shows that LiSP-XK is 10%, 28.4%, 14.1% and 16.5% better efficient in communication cost as compared to the schemes mentioned in [13] and [57]–[59], respectively. The use of light-weight public-key cryptography, Keccak hash, and reduced the size of random number and timestamp and has intensified the results maintaining the suitable security parameter. To compute the complexity, we first define the operational time parameters as shown in Table 4. These time parameters are measured on the hardware configuration mentioned in Table 1 and used for all the schemes in comparison. The unit for this metric is milliseconds (ms). Note that the other existing schemes use different hashing and encryption-decryption processes leading to the increased time of operations. The comparison of the computation complexity is shown in Table 5. The observations and calculations denote that, our LiSP-XK is less complex by 39.2%, 26.1%, 39.9% and 25.6% as compared to the works in [13] and [57]–[59], respectively. This proves that LiSP-XK is lighter than the existing schemes and thus applicable for resource-constrained environments.

TABLE 3. Communication cost comparison.

Protocol	Total communication cost (in bits)
LiSP-XK	2336
LiSP [13]	2600
Liu et al. [57]	3267
Zhou et al. [58]	2720
Elkhalil et al. [59]	2800

The increasing demand of IoT applications also emphasizes energy consumption and memory consumption.

TABLE 4. Time for related operation.

Operation	Symbol	Average time
Addition and multiplication	T_o	10.405
Hashing	T_h	0.165
Encryption-decryption	T_{ed}	0.780

TABLE 5. Computation costs comparison.

Protocol	Total cost	Approximately time (ms)
LiSP-XK	$6T_o + 6T_h + 2T_{ed}$	64.98
LiSP [13]	$8T_h + 10T_o + 2T_{ed}$	106.93
Liu et al. [57]	$8T_h + 8T_o + 4T_{ed}$	88.04
Zhou et al. [58]	$6T_h + 10T_o + 4T_{ed}$	108.16
Elkhalil et al. [59]	$6T_h + 8T_o + 4T_{ed}$	87.35

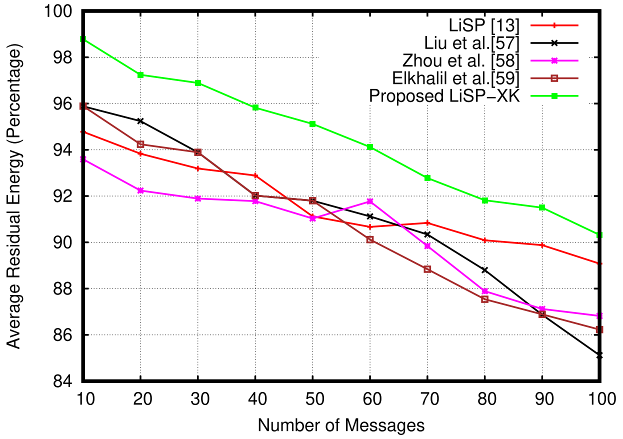

FIGURE 3. Comparison of residual energy parameter.

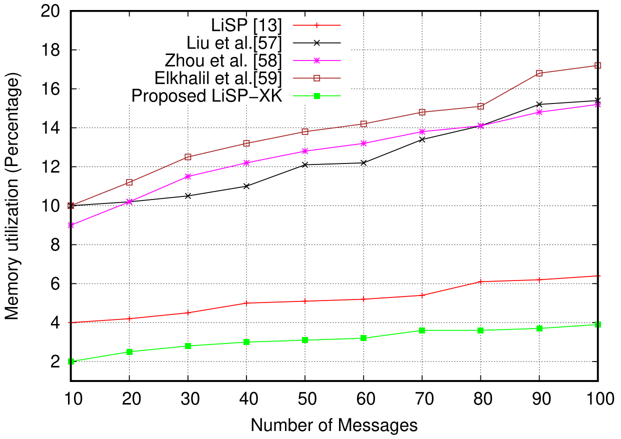

FIGURE 4. Comparison of memory consumption.

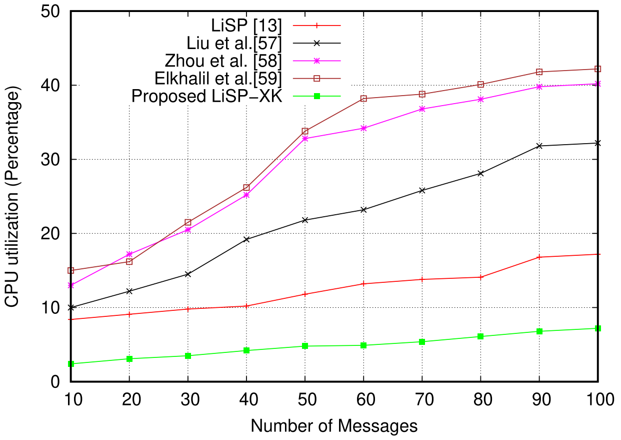

FIGURE 5. Comparison of CPU utilization.

The reason for these considerations is due to the resource-constrained devices with low memory sizes. Therefore, we require to check the energy consumption and memory utilization ratio for LiSP-XK and compare it with the other existing approaches. To measure the energy efficiency, we use the residual energy as mentioned in [13]. The formula used for this is given as:

$$Energy_{res} = \frac{\sum_{i=1}^{n} Energy_{init} - Energy_{T}}{number\ of\ iterations} \tag{1}$$

where, $Energy_{init}$ is the initial energy of the nodes and $Energy_T$ is the energy of the nodes after the time-period T. The comparative study of energy has been shown in Figure 3. It shows that LiSP-XK consumes 33.5% less energy. Note that, it has been calculated as residual energy for overall nodes and then the average has been counted for the comparison. We can also see that LiSP-XK stands apart from the other signcryption approaches in comparison. As we have used Elli and Keccak - the two light-weight cryptographic methods - computation is less complex and therefore, energy consumption is less as compared to others. Resource consumption is an important factor in energy constrained devices. We consider here two types of resources to measure: memory consumption and CPU utilization. First, we observe the memory consumption behaviour of the systems by checking the utilization of memory space for varying the number of message transfers from 10 to 100. The results are shown in Figure 4 in terms of memory consumption out of available memory of the nodes. Figure 4 shows that LiSP-XK is 30% better in memory utilization. The results also show that LiSP-XK's memory utilization is better than the other schemes. The use of Keccak reduces memory consumption and thus, our proposed scheme obtains efficiency in this with benefits.

Next, we measure the CPU utilization. We show the comparative outcomes in percentage in Figure 5. It shows that LiSP-XK provides 34.3% reduced CPU utilization which is beneficial for resource constrained applications. As, the Keccack and Elli both are light weight, these two helps to make LiSP-XK to become more light weight as compared to LiSP. Moreover, the reduced number of steps in computing and efficient use of the key values are add on for this benefit.

C. SECURITY REQUIREMENTS

In this section, we analyze the security notions of LiSP-XK on perspectives of attack handle and the requirements:

- **Forward Secrecy:** Forward secrecy of an authentication scheme assurances that the session keys are not vulnerable to be compromised, even if long-term secrets used in the session key exchange are compromised. In LiSP-XK, the secret seed on the elliptic curve and λ both are randomly chosen. Therefore, if any one of them is compromised, the other parameter can still be secure. Moreover, to provide further security, both the elliptic curve seed and the random λ can be updated with extra computation cost if required.

- **Backward Secrecy:** The use of random numbers (salt) and hash helps in this process. As if the adversary gets knowledge about the subset or partial key, the private key derivation is quite unsuccessful and thus, LiSP-XK is secure.
- **Handling Impersonation:** Suppose Eve impersonates as an authorized entity in the network. To succeed with her intention, a chooses randomly value of λ'. To send the message, $R = \lambda^{-1}(N_{S_{pvr}} + q)$ needs to be calculated by $\mathcal{A}$. If the receiver does not identify the corresponding public-key, it will be granted as an invalid message following Algorithm 2.
- **Handling Replay:** We use the time-stamp validity check for the receiver to process the message. This is able to handle stale messages or replay messages.

V. CONCLUSION

In this paper, we extend the LiSP signcryption to LiSP-XK signcryption. We use Elli and Keccak as the light-weight cryptographic modules to obtain the desired functionality of the light-weight signcryption. We study the experimental outcomes of LiSP-XK and compare the results with state-of-the-art schemes. It shows that LiSP-XK is overall 35% better in efficiency as compared to the other signcryption approaches. The measured light-weight in terms of reduced complexity, communication cost, energy and memory consumption makes our LiSP-XK suitable for resource-constrained environments such as IIoTs, WSNs and others. In future, we will try to work on a suitable distributed key generation mechanism for more effective results.

REFERENCES

[1] F. H. Al-Naji and R. Zagrouba, "A survey on continuous authentication methods in Internet of Things environment," *Comput. Commun.*, vol. 163, pp. 109–133, Nov. 2020.

[2] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A survey on privacy and security of Internet of Things," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100312.

[3] S.-J. Horng, S.-F. Tzeng, Y. Pan, P. Fan, X. Wang, T. Li, and M. K. Khan, "b-SPECS+: Batch verification for secure pseudonymous authentication in VANET," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1860–1875, Nov. 2013.

[4] W. Z. Khan, M. Y. Aalsalem, M. K. Khan, M. S. Hossain, and M. Atiquzzaman, "A reliable Internet of Things based architecture for oil and gas industry," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2017, pp. 705–710.

[5] P. K. Malik, R. Sharma, R. Singh, A. Gehlot, S. C. Satapathy, W. S. Alnumay, D. Pelusi, U. Ghosh, and J. Nayak, "Industrial Internet of Things and its applications in industry 4.0: State of the art," *Comput. Commun.*, vol. 166, pp. 125–139, Jan. 2021.

[6] S.-F. Tzeng, S.-J. Horng, T. Li, X. Wang, P.-H. Huang, and M. K. Khan, "Enhancing security and privacy for identity-based batch verification scheme in VANETs," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3235–3248, Apr. 2017.

[7] M. Mahbub, "Progressive researches on IoT security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics," *J. Netw. Comput. Appl.*, vol. 168, Oct. 2020, Art. no. 102761.

[8] M. Henriquez. *Security: The Top 10 Data Breaches of 2020*. Accessed: Dec. 12, 2020. [Online]. Available: https://www.securitymagazine.com/articles/94076-the-top-10-data-breaches-of-2020

[9] D. He, H. Wang, M. K. Khan, and L. Wang, "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," *IET Commun.*, vol. 10, no. 14, pp. 1795–1802, Sep. 2016.

[10] V. Odelu, A. K. Das, M. K. Khan, K. R. Choo, and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, 2017.

[11] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[12] M. S. Virat, S. M. Bindu, B. Aishwarya, B. N. Dhanush, and M. R. Kounte, "Security and privacy challenges in Internet of Things," in *Proc. 2nd Int. Conf. Trends Electron. Informat. (ICOEI)*, May 2018, pp. 454–460.

[13] A. Kumar, R. Saha, M. Alazab, and G. Kumar, "A lightweight signcryption method for perception layer in Internet-of-Things," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102662.

[14] B. Butani, P. K. Shukla, and S. Silakari, "An exhaustive survey on physical node capture attack in WSN," *Int. J. Comput. Appl.*, vol. 95, no. 3, pp. 32–39, Jun. 2014.

[15] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption)," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, Aug. 1997, pp. 165–179.

[16] M. K. Khan, "Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world," *IETE Tech. Rev.*, vol. 26, no. 3, pp. 191–195, 2009.

[17] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks,'" *Sensors*, vol. 10, no. 3, pp. 2450–2459, Mar. 2010.

[18] J. Fu, C. Wu, X. Chen, R. Fan, and L. Ping, "Scalable pseudo random RFID private mutual authentication," in *Proc. 2nd Int. Conf. Comput. Eng. Technol.*, vol. 7, Apr. 2010, pp. V7-497–V7-500.

[19] Y.-Q. Gui and J. Zhang, "A new authentication RFID protocol with ownership transfer," in *Proc. Int. Conf. ICT Converg. (ICTC)*, Oct. 2013, pp. 359–364.

[20] S. R. Moosavi, E. Nigussie, S. Virtanen, and J. Isoaho, "An elliptic curve-based mutual authentication scheme for RFID implant systems," *Procedia Comput. Sci.*, vol. 32, pp. 198–206, Jan. 2014.

[21] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 2728–2733.

[22] M. Chen and S. Chen, "An efficient anonymous authentication protocol for RFID systems using dynamic tokens," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, Jun. 2015, pp. 756–757.

[23] P. Gope and T. Hwang, "A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system," *Comput. Secur.*, vol. 55, pp. 271–280, Nov. 2015.

[24] P. Dass and H. Om, "A secure authentication scheme for RFID systems," *Procedia Comput. Sci.*, vol. 78, pp. 100–106, Jan. 2016.

[25] R. Amin, S. K. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[26] S. Arasteh, S. F. Aghili, and H. Mala, "A new lightweight authentication and key agreement protocol for Internet of Things," in *Proc. 13th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2016, pp. 52–59.

[27] M. T. Hammi, E. Livolant, P. Bellot, A. Serhrouchni, and P. Minet, "A lightweight mutual authentication protocol for the IoT," in *Proc. Int. Conf. Mobile Wireless Technol.* Singapore: Springer, Jun. 2017, pp. 3–12.

[28] P. K. Dhillon and S. Kalra, "A lightweight biometrics based remote user authentication scheme for IoT services," *J. Inf. Secur. Appl.*, vol. 34, pp. 255–270, Jun. 2017.

[29] M. Lavanya and V. Natarajan, "Lightweight key agreement protocol for IoT based on IKEv2," *Comput. Electr. Eng.*, vol. 64, pp. 580–594, Nov. 2017.

[30] C. J. F. Cremers, *Scyther: Semantics and Verification of Security Protocols*. Eindhoven, The Netherlands: Eindhoven University of Technology, 2006.

[31] A. Tewari and B. B. Gupta, "A robust anonymity preserving authentication protocol for IoT devices," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–5.

[32] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, Jan. 2018.

[33] W. Feng, Y. Qin, S. Zhao, and D. Feng, "AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS," *Comput. Netw.*, vol. 134, pp. 167–182, Apr. 2018.

[34] F. Nikkhah and M. Safkhani, "LAPCHS: A lightweight authentication protocol for cloud-based health-care systems," *Comput. Netw.*, vol. 187, Mar. 2021, Art. no. 107833.

[35] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. A. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107333.

[36] S. F. Aghili, H. Mala, P. Kaliyar, and M. Conti, "SecLAP: Secure and lightweight RFID authentication protocol for medical IoT," *Future Gener. Comput. Syst.*, vol. 101, pp. 621–634, Dec. 2019.

[37] I. A. Kamil and S. O. Ogundoyin, "A lightweight mutual authentication and key agreement protocol for remote surgery application in tactile internet environment," *Comput. Commun.*, vol. 170, pp. 1–18, Mar. 2021.

[38] Minahil, M. F. Ayub, K. Mahmood, S. Kumari, and A. K. Sangaiah, "Lightweight authentication protocol for e-health clouds in IoT-based applications through 5G technology," *Digit. Commun. Netw.*, vol. 7, no. 2, pp. 235–244, May 2021.

[39] L. Ning, Y. Ali, H. Ke, S. Nazir, and Z. Huanli, "A hybrid MCDM approach of selecting lightweight cryptographic cipher based on ISO and NIST lightweight cryptography security requirements for internet of health things," *IEEE Access*, vol. 8, pp. 220165–220187, 2020.

[40] X. Tan, J. Zhang, Y. Zhang, Z. Qin, Y. Ding, and X. Wang, "A PUF-based and cloud-assisted lightweight authentication for multi-hop body area network," *Tsinghua Sci. Technol.*, vol. 26, no. 1, pp. 36–47, Feb. 2021.

[41] Z. U. Rehman, S. Altaf, and S. Iqbal, "An efficient lightweight key agreement and authentication scheme for WBAN," *IEEE Access*, vol. 8, pp. 175385–175397, 2020.

[42] A. A. Khan, V. Kumar, M. Ahmad, and S. Rana, "LAKAF: Lightweight authentication and key agreement framework for smart grid network," *J. Syst. Archit.*, vol. 116, Jun. 2021, Art. no. 102053.

[43] K. Wu, R. Cheng, W. Cui, and W. Li, "A lightweight SM2-based security authentication scheme for smart grids," *Alexandria Eng. J.*, vol. 60, no. 1, pp. 435–446, Feb. 2021.

[44] D. Sadhukhan, S. Ray, M. S. Obaidat, and M. Dasgupta, "A secure and privacy preserving lightweight authentication scheme for smart-grid communication using elliptic curve cryptography," *J. Syst. Archit.*, vol. 114, Mar. 2021, Art. no. 101938.

[45] L. Zhang, L. Zhao, S. Yin, C.-H. Chi, R. Liu, and Y. Zhang, "A lightweight authentication scheme with privacy protection for smart grid communications," *Future Gener. Comput. Syst.*, vol. 100, pp. 770–778, Nov. 2019.

[46] B. D. Deebak and F. Al-Turjman, "Lightweight authentication for IoT/cloud-based forensics in intelligent data computing," *Future Gener. Comput. Syst.*, vol. 116, pp. 406–425, Mar. 2021.

[47] I. U. Haq, J. Wang, and Y. Zhu, "Secure two-factor light-weight authentication protocol using self-certified public-key cryptography for multi-server 5G networks," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102660.

[48] M. Wazid, A. K. Das, K. V. Bhat, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102496.

[49] B. Li, W. Liu, and L. Wang, "Efficient and lightweight batch authentication for large-scale RFID systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1272–1275, Aug. 2019.

[50] Y. Aydin, G. K. Kurt, E. Ozdemir, and H. Yanikomeroglu, "A flexible and lightweight group authentication scheme," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10277–10287, Oct. 2020.

[51] Z. Liu, C. Guo, and B. Wang, "A physically secure, lightweight three-factor and anonymous user authentication protocol for IoT," *IEEE Access*, vol. 8, pp. 195914–195928, 2020.

[52] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, Feb. 2019.

[53] J. Cao, Z. Yan, R. Ma, Y. Zhang, Y. Fu, and H. Li, "LSAA: A lightweight and secure access authentication scheme for both UE and mMTC devices in 5G networks," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5329–5344, Jun. 2020.

[54] P.-Y. Ting, J.-L. Tsai, and T.-S. Wu, "Signcryption method suitable for low-power IoT devices in a wireless sensor network," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2385–2394, Sep. 2018.

[55] V. Rajasekar, S. Varadhaganapathy, K. Sathya, and J. Premalatha, "An efficient lightweight cryptographic scheme of signcryption based on hyper-elliptic curve," in *Proc. 3rd Int. Conf. Recent Adv. Inf. Technol. (RAIT)*, Mar. 2016, pp. 394–397.

[56] A. K. Singh and K. S. Vaisla, "A lightweight signcryption scheme based on elliptic curve cryptography," in *Proc. 1st Int. Conf. Adv. Comput. Commun. Eng. (ICACCE)*, vol. 1, 2014, pp. 7–10.

[57] X. Liu, Z. Wang, Y. Ye, and F. Li, "An efficient and practical certificateless signcryption scheme for wireless body area networks," *Comput. Commun.*, vol. 162, pp. 169–178, Oct. 2020.

[58] Y. Zhou, Y. Xu, Z. Qiao, B. Yang, and M. Zhang, "Continuous leakage-resilient certificate-based signcryption scheme and application in cloud computing," *Theor. Comput. Sci.*, vol. 860, pp. 1–22, Mar. 2021.

[59] A. Elkhalil, J. Zhang, R. Elhabob, and N. Eltayieb, "An efficient signcryption of heterogeneous systems for internet of vehicles," *J. Syst. Archit.*, vol. 113, Feb. 2021, Art. no. 101885.

[60] H. Xiong, Y. Hou, X. Huang, and Y. Zhao, "Secure message classification services through identity-based signcryption with equality test towards the internet of vehicles," *Veh. Commun.*, vol. 26, Dec. 2020, Art. no. 100264.

[61] S. Hussain, I. Ullah, H. Khattak, M. A. Khan, C.-M. Chen, and S. Kumari, "A lightweight and provable secure identity-based generalized proxy signcryption (IBGPS) scheme for industrial Internet of Things (IIoT)," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102625.

[62] M. Nikravan, A. Movaghar, and M. Hosseinzadeh, "A lightweight signcryption scheme for defense against fragment duplication attack in the 6LoWPAN networks," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 1, pp. 209–226, Jan. 2019.

[63] V. Rao and K. V. Prema, "Light-weight hashing method for user authentication in Internet-of-Things," *Ad Hoc Netw.*, vol. 89, pp. 97–106, Jun. 2019.

[64] N. Eltayieb, R. Elhabob, A. Hassan, and F. Li, "A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud," *J. Syst. Archit.*, vol. 102, Jan. 2020, Art. no. 101653.

[65] S. Mella, J. Daemen, and G. Van Assche, "New techniques for trail bounds and application to differential trails in Keccak," *IACR Trans. Symmetric Cryptol.*, pp. 329–357, Mar. 2017.

[66] R. Saha and G. Geetha, "Symmetric random function generator (SRFG): A novel cryptographic primitive for designing fast and robust algorithms," *Chaos, Solitons Fractals*, vol. 104, pp. 371–377, Nov. 2017.

[67] ELLI. *Light Weight Public Key*. Accessed: Dec. 02, 2020. [Online]. Available: https://asecuritysite.com/encryption/elli

[68] W. J. Buchanan, S. Li, and R. Asif, "Lightweight cryptography methods," *J. Cyber Secur. Technol.*, vol. 1, nos. 3–4, pp. 187–201, 2017.